DOCUMENT RESUME

ED 281 485 IR 012 601

AUTHOR Mandinach, E.; And Others
TITLE Review of Research on the Cognitive Effects of Computer-Assisted Learning.
INSTITUTION California Univ., Berkeley. Lawrence Hall of Science.; Far West Lab. for Educational Research and Development, Berkeley, Calif.
PUB DATE Jul 83
NOTE 88p.
PUB TYPE Information Analyses (070) -- Reports - Descriptive (141)

EDRS PRICE MF01/PC04 Plus Postage.
DESCRIPTORS Aptitude Treatment Interaction; Artificial Intelligence; *Cognitive Processes; Cognitive Structures; Cognitive Style; *Computer Assisted Instruction; Computer Simulation; *Computer Software; Educational Games; Elementary Secondary Education; Epistemology; *Learning Theories; Literature Reviews; Microcomputers; Problem Solving; Research Design; *Research Utilization

ABSTRACT

This review of the research on the cognitive effects of computer-assisted instruction begins with an overview of the ACCCEL (Assessing Cognitive Consequences of Computer Environments for Learning) research program at the University of California at Berkeley, which consists of several interrelated studies examining the acquisition of such higher cognitive skills as strategic planning knowledge, inductive reasoning, and deductive reasoning in computer learning environments. The context and theoretical issues in computer environments for learning are also discussed, including claims about such environments, distinctive features of these environments, and pedagogical and learning theories related to software design. Among the subject areas covered in this comprehensive literature review are: (1) individual differences in aptitude and ability and ATI (aptitude treatment interaction) research, including cognitive styles and affective factors; (2) problem solving and knowledge states; (3) instructional psychology; (4) planning; and (5) computer-assisted instruction, computer tutors, and artificial intelligence. Characteristics of software selected for study--WUMPUS, Creative Play, Rocky's Boots, Yellow Light, and Pattern Discovery--are described, and a discussion of the implications of the research reviewed for the design and implementation of future research concludes the report. An 11-page bibliography is provided. (MES)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED281485

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
X This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

REVIEW OF RESEARCH ON THE COGNITIVE EFFECTS OF COMPUTER-ASSISTED LEARNING

by

E. Mandinach
C. Fisher

with

N. Burbules
J. Dalbey
M. Linn
P. Reese

July, 1983

ACCCEL: Assessing Cognitive Consequences of Computer Environments for Learning

a joint project of

Lawrence Hall of Science
University of California

and

The Far West Laboratory for
Educational Research & Development

IR012601

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Marcia Linn

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

BEST COPY AVAILABLE

OVERVIEW OF THE ACCCEL RESEARCH PROGRAM

Extensive use of microcomputers in schools, homes, banks, supermarkets and many other settings make computer learning an important and immediate issue in education. If many of the claims about computer learning are true, the use of microcomputers may herald profound changes in the structure of education. It is already clear that microcomputers are and will be used for educational purposes. Perhaps the important questions at this point are, what are the cognitive demands placed on learners, and what are the consequences of learning activities conducted in a computer learning environment?

There are several classes of cognitive consequences that may result from computer learning experiences. Two of these classes are (a) acquisition of subject matter knowledge, and (b) acquisition of thinking and logical reasoning skills (higher cognitive skills). Although the learning of subject matter knowledge is clearly important, it is not addressed directly in this research program. Our focus is on the acquisition of higher cognitive skills as a result of learner activity in computer learning environments. The higher cognitive skills in which we are most interested are (a) strategic planning knowledge, (b) inductive reasoning, and (c) deductive reasoning.

What is it about computer learning environments that can be expected to influence the acquisition of higher cognitive skills? In what ways do computer learning environments differ from other learning environments that could lead to changes in higher cognitive

skills of learners?

Our analysis of computer learning environments proposes six characteristics of these environments which, by theory or previous research, appear to be related to acquisition of higher cognitive skills. The characteristics are: (a) interactiveness; (b) precision; (c) consistency; (d) challenge; (e) complexity; and (f) provision for multiple solutions.

The impact of these characteristics of computer learning environments on higher cognitive skills is expected to depend upon two classes of factors. First, learners differ in aptitude and therefore may acquire differential skills and knowledge from computer learning experiences. Second, learners who differ in cognitive style variables also may acquire higher cognitive skills in different manners or in different degrees as a result of experience in computer learning environments. The focus of the research is refined further to investigate the acquisition of higher cognitive skills as a result of learner activities in computer learning environments, taking into account learner differences in aptitude and cognitive style variables.

Thus, the central framework of the research program examines how learners who differ in aptitude and cognitive styles, and with respect to experience in computer learning environments, differentially acquire higher congitive skills and knowledge.

Within this framework, we have identified three classes of computer learning environments that seem to warrant investigation. The three classes are: (a) computer programming environments (the learner uses a "language" to program the computer); (b) educational simulations and games (the learner "controls" various elements in

order to attain single or multiple goals); and (c) subject matter presentation and practice routines. These environments differ in the extent to which one or more of the six characteristics of computer learning environments are manifested, and therefore in the kind and extent of influence learner experience in these environments may have on higher cognitive skills acquisition.

The first two classes of environments, computer programming and educational simutations and games, make extensive use of the six characteristics of computer learning environments and therefore may have substantial impact on higher cognitive skills. The third class (subject matter presentation and practice routines) is a direct translation of traditional instructional procedures and may be less likely to enhance higher cognitive skills. For this reason, we are considering computer programming and educational simulations and games as the two classes of computer learning environments most relevant to the pursuit of the specified research objectives.

Regardless of the class of computer learning environment, many other treatment conditions are likely to affect higher cognitive skills. For example, the degree to which strategic planning knowledge (or structured programming conepts) is made explicit may be a critical dimension in determining what and how much is learned. Therefore, we systematically include instructional treatments at two levels of explicitness in each of the computer learning environments proposed for study.

The current phase of the ACCCEL research program consists of eight interrelated studies to be conducted between January 1983 and January 1985. The studies are divided into two groups: Series A

studies and Series B studies.

The studies in Series A examine the demands and higher cognitive consequences of learning computer programming. Series B studies examine the higher cognitive consequences of learner activities using educational simulations and games. Both series have been designed carefully to include the same (or similar) aptitude, cognitive style, and instructional treatment variables, as well as similar outcome variables (on both near and far transfer tasks). In these interrelated studies, we examine the acquisition of higher cognitive skills for learners of different: (a) ages (ten-year-olds and junior high school-age students); (b) socioeconomic background (low and medium levels of SES); (c) computer-related experience (no experience and previous instruction using computers); (d) gender (male and female); and (e) previous achievement level. We also examine the acquisition of higher cognitive skills when the computer environment is housed in different settings (schools, museums, and other settings). In addition, within the Series A studies, we will explore three programming languages (BASIC, Pascal, and Spider); within the Series B studies we will examine the use of several educational simulations and games (Creative Play, WUMPUS, Rocky's Boots, and the Yellow Light Problem).

In order to incorporate each of the factors discussed above in the overall design of the ACCCEL research program, two main studies and a series of smaller studies will be conducted. The smaller studies extend the main studies by examining carefully chosen treatment groups representing combinations of the factors discussed above.

More extensive information on the background and rationale for the research program is provided in the research proposal.[1] A list of documents associated with the ACCCEL research program is included as Table 1.

The current document presents a literature review on the demands and consequences of learning from computer software. We begin with a discussion of the context and theoretical issues in computer environments for learning. These issues involve the claims, distinctive features, and pedagogical theories related to the design of and learning from software. Second, we review several bodies of literature that are relevant to computer assisted learning. Third, we consider the characteristics of software that exemplify cognitively demanding computer tasks. Finally, we conclude with a discussion of the implications of relevant issues to the design and implementation of future research.

M. Linn
C. Fisher

---

[1] Information on the ACCCEL research proposal is available from the ACCCEL program offices at Lawrence Hall of Science, University of California, Berkeley, California or Far West Laboratory for Educational Research and Development, San Francisco, California.

TABLE 1

Reports and Due Dates for the ACCCEL Research Program

TENTATIVE REPORT TITLE	TASK	DUE DATE	
		DRAFT	FINAL
1. Literature Review on Cognitive Demands and Requirements of Computer Programming	1	Mar. 31, 1983	Oct. 30, 1983
2. Research Design for Studies on the Acquisition of Higher Cognitive Skills from Learning Computer Programming (Series A Studies)	2-3	Jul. 31, 1983	Aug. 31, 1983
3. Final Report on the Requirements for Learning from Computer Programming	2	--	Jun. 30, 1984
4. Final Report on the Acquisition of Higher Cognitive Skills from Learning Computer Programming	4	--	Sep. 30, 1984
5. Cognitive Requirements and Benefits of Learning Computer Programming	5	Dec. 15, 1984	Jan. 9, 1985
6. Selection of software for Studies of Acquisition of Higher Cognitive Skills from Computer-Assisted Learning	6	--	Apr. 15, 1983
7. Review of Research on the Cognitive Effects of Computer-Assisted Learning	7	Jun. 30, 1983	Jan. 30, 1984
8. Research Design for Studies of Acquisition of Higher Cognitive Skills from Computer-Assisted Learning	9	Studies B1 & B4 Studies B2 & B3	Mar. 15, 1983 Nov. 30, 1983
9. Final Report on Acquisition of Higher Cognitive Skills from Computer-Assisted Learning	10	--	Jan. 9, 1985

TABLE OF CONTENTS

CHAPTER ONE: CONTEXT OF AND THEORETICAL ISSUES IN COMPUTER ENVIRONMENTS FOR LEARNING 1

Introduction 1
Claims About Computer Environments for Learning 2
Distinctive Features of Computer Environments for Learning 3
Pedagogical and Learning Theories Related to Software Design 6

CHAPTER TWO: REVIEW OF RELEVANT RESEARCH 11

Problem Situation 11
Definition of the Research Problem 13
Individual Differences 13
Aptitude and Ability Differences and and ATI Research 13
Cognitive Styles and Affective Factors 19
Instructional Psychology 23
Transfer 23
Rule and Discovery Learning 24
Learning from Examples and Induction 25
Problem Solving and Knowledge States 28
Training Problem Solving Skills 28
Ability and Strategic Differences in Problem Solving Performance 30
Knowledge States 31
Planning 33
Planning in Aritificial Intelligence 34
Human Planning Processes: The Opportunistic Model 35
Computer Applications and CAI 36
Computer Tutors and Coaches 37
Educational Games and Simulations 38
Issues Affecting Implementation of CAI 39

CHAPTER THREE: RESEARCH APPLICATIONS: CHARACTERISTICS OF SOFTWARE 43

Introduction 43
Proposed Courseware Dimensions 43
Applicability, Structures, and Instructional Intent 44
General Dimensions of Software 45
Rationale for Selected Software 46
Overview 46

Table of Contents (Continued) Page

CHAPTER THREE (continued)

Study B-1: WUMPUS. 47
WUMPUS 48
Instructional WUMPUS 51
Study B-2: Rocky's Boots 52
Study B-3: Creative Play 53
Study B-4: Yellow Light. 54
Study B-5: Pattern Discovery 62

CHAPTER FOUR: IMPLICATIONS FOR RESEARCH 65

References 69

CHAPTER ONE

Context of and Theoretical Issues in Computer Environments for Learning

Introduction

Computers and computer assisted learning are commonly found in American schools. Elementary, junior high, and high schools alike are moving quickly to acquire hardware, select and use software, develop sequences of activities for students, and train instructors to implement the new activities. Sixty-eight percent of a nationally representative sample of public and non-public elementary and secondary schools had one or more microcomputers for use of teachers or students by January 1983 (Center for Social Organization of Schools, 1983). Whether or not computers and computer assisted instruction will have an important impact on schooling is no longer a question. The key issue now is to explore the consequences for student learning.

Many expectations have been expressed about the effects of this technological move on both student learning and the structure of schools. Although the most frequent use of computers for instructional purposes still appears to be restricted to drill and practice situations, there are increasing numbers of courses on programming, word processing, and incorporation of simulations and games into standard subject area courses. Unlike the large, costly and complex computer assisted instructional systems introduced into education approximately 20 years ago, microcomputers of today are small, relatively inexpensive, and remarkably flexible machines. Their positive impact on schools and on student learning has been often predicted and as miniaturization, cost reduction, and flexibility continue to increase public expectations also continue to rise. Grand claims have been made about the powerful effects of computers in schools. However, at the present time, these claims remain unsubstantiated by experimental evidence. Research is necessary to examine the extent to which such claims are valid representations of the demands and consequences of computer environments for learning. The overall goal of this review is to examine the major areas of work on the cognitive prerequisites and consequences of computer environments for learning. Although the review often includes material that is relevant to learning computer programs, the main focus is on the cognitive consequences of already programmed material, especially cognitively demanding computer simulations and games. (The ACCCEL project has reviewed the cognitive demands and consequences of computer programming in another document by Dalbey & Linn, 1983.)

Before presenting our review, we outline the range of expectations that have been expressed about the outcomes of computer

environments for learning; indicate our primary interest within this range; outline six characteristics of computer environments for learning that are likely to facilitate certain types of student learning; and briefly comment on theories of learning that may have particular relevance in the analysis of computer environments for learning.

Although a broad variety of effects may result from the use of computer simulations and games, our focus is on the cognitive consequences, more specifically on the acquisitoin of problem solving skills. Our focus here is on the cognitive consequences of learning from computer games and simulations. A broad variety of effects may result from the use of this class of software. However, our primary interest is on the acquisition of problem solving skills that result from interaction with such educational games and simulations. First, it is necessary to consider a number of issues that relate to and influence computer environments for learning. First, we discuss the distinctive features of the learning environment and some of the pedagogical and learning theories relevant to software design. We then review several areas of research that relate to computer learning. Third, we examine potential research applications and identify dimensions of software that have been selected for study. Finally, we conclude with a discussion of possible implications for future research on the effects of computer environments for learning.

Claims About Computer Environments for Learning

What are the expectations that have been expressed about computers and computer assisted instruction in schools? Some arguments have suggested that, with the appropriate software, computer assisted instruction may enhance the development of higher cognitive skills of students. For example, the opportunity for students to interact with well-designed simulations and games may lead to better problem solving skills. Others argue that the microcomputer will make possible new instructional arrangements that should lead directly to increased productivity of schools and increased student learning. Some classroom management activities may be computerized thereby freeing teachers to reallocate their time to instructional activities. Computer assisted instruction may allow individual students to work on individualized tasks and have those tasks specially designed to take advantage of individual learner characteristics and learning style variables. Improvements in instruction resulting from computer assisted instruction may allow more equitable access to knowledge and skills by members of special populations (handicapped students, students with limited proficiency in English, or students from extreme ability groups). Sophisticated computer assisted instruction techniques may allow appropriately differentiated instruction for females and males or for students with widely differeing social backgrounds and thereby reduce the differences that arise between these groups under current schooling conditions.

Extensive use of computer assisted instruction may change social interaction patterns in schools and thereby affect student learning outcomes. For example, increased student autonomy in terms of task definition, duration, and sequencing could increase learner control over the learning environment and thus facilitate learning how-to-learn on other higher cognitive skills that, under current school practice, are often unaffected. Computer environments for learning often encourage peer tutoring among students in a manner that is relatively uncommon in schools. These changes in the roles of students during instruction and concomitant changes in teacher role may change student perceptions of self and indirectly affect student learning.

Still other arguments regarding the use of comuter assisted instruction claim that experience with computers will add to the range of vocational choices open to students. That is, early introduction and familiarity with computers and computing will ultimately enable some students to pursue careers in technological fields. The issue of vocational opportunity is not restricted to concerns for preparation of relatively esoteric specialists for technologically sophisticated occupations. School experience in computer environments for learning may enable much broader segments of the population to consider and pursue the wide varieties of occupations that will inevitably incorporate information processing machines and thus require some degree of computer familiarity and skill as a central or peripheral aspect of the job.

Although additional arguments have been advanced for the implementation of computer assisted instruction in schools, it is already clear that the range of hypothesized effects is very large and, if found, the effects include critically important variables. What is it about computer learning environments that could possibly give rise to such effects?

Distinctive Features of Computer Environments for Learning

Certain features of the computer learning environment distinguish it from other learning environments and make it a potential facilitator for the acquisition of a wide variety of cognitive and other related skills. Six key features have been identified: (a) interactiveness, (b) precision, (c) consistency, (d) challenge, (e) complexity, and (f) provision for multiple solutions. Each of these features is examined in turn.

First, computer environments for learning are interactive. Whether one is using a computer simulation, playing a computer game, or developing a computer program, there is a cyclical process of providing information to and receiving feedback from the computer. The time required for one cycle of the process varies considerably. For batch processing of computer programs, the typical response-feedback cycle may be as long as one day; for interactive computer games, the response-feedback cycle can test the limits of human response time. Although the response-

feedback cycle time can vary greatly, all computer environments for learning are characterized by interactiveness. To proceed with any computer task, the learner must respond actively. That is, the learner must submit another program, initiate another keystroke, or move the paddle control. The learner cannot be passive in the environment, at least in comparison to the passivity characteristic of many more traditional learning enviroments. The requirement that the learner respond periodically (sometimes very quickly) insures that some level of cognitive processing must occur or the interaction will come to a halt. In this way, computer environments for learning encourage cognitive activity, and this activity, in turn, is expected to have consequences for cognitive performance on tasks that do not involve a computer. Of course, the quality of the cognitive activity generated in computer environments for learning is dependent upon the software or programming task that is used. Interactiveness is not unqiue to comuter learning environments. However, the degree to which computerlearner interactions can be controlled and potentially used for increasing learning efficiency is an important characteristic of computer environments for learning.

Second, computer environments for learning are precise. To communicate with the machine, the learner must specify his message completely. In general, the computer does not make inferences, "read between the lines," or otherwise use implicit knowledge to interpret the desires of the learner. Computer environments for learning demand precision. Learners are required to differentiate among items that previously were undifferentiated, to "think through" aspects of a problem that previously were unclear, or to articulate one's tacit knowledge. In this manner, the preciseness of the computer environments for learning encourages relatively complete specification of messages. To the extent that the preciseness of computer environments for learning fosters these kinds of activities, then we can expect experience in computer environments for learning to have cognitive consequences for learning tasks that do not involve computers.

Third, computer environments for learning are consistent. This consistence is manifested in several ways. If the learner provides information to the computer (in the form of a program, or a series of keystrokes), the computer will respond in a specific and concrete manner. If the learner repeats his action (resubmits the program or repeats the same series of keystrokes), the computer will give the exact same response as was given on the first trial. If the learner does the same thing, the computer will do the same thing. Thus the computer gives consistent responses (feedback) from one session to another.

Computer environments for learning are consistent in a second important way. The computer will provide a consistent response (to the same input) regardless of who the learner/programmer is. An obese or a thin person, a tall or short person, a native English-speaker or native Spanish-speaker, a black-skinned or brown-skinned person all will get the same response from the computer, provided the same information was "fed in." That is, the feedback provided to learners does not vary with "irrelevant" characteristics of the learner. These

more or less obvious points about the consistency of computer environments for learning have important implications for what and how much students learn. Since the feedback in computer environments for learning is consistent, relationships among the learning actions and the computer's feedback should be easier to identify, compared to other environments where the feedback is less than consistent. Consistent feedback reduces the processing load, because the learner has to deal with less ambiguity since the computer's precision can eliminate the extraneous variables, and, other things being equal, should result in more efficient learning for most learners.

Fourth, the computer provides a challenging and motivational environment in which to learn. Many teachers have observed the motivational qualities of the computer in the classroom. Malone (1981) categorized the qualities of computer games that make them intrinsically motivating. These qualities are challenge, fantasy, and curiosity. Optimal levels of challenge and curiosity depend on goals with uncertain outcomes that maintain motivation and stimulate the student to respond to feedback and complete partial cognitive structures. Uncertain outcomes stimulate challenge while controlled feedback allows students to form consistent expectations, not only of their own performance but also the computer's actions. Burton and Brown (1979) defend the use of games and simulations in learning environments. They state that games are good problem solving environments that stimulate explanation, discovery learning, and formulation of strategies and knowledge structures.

Fifth, computer environments for learning are complex. Computers, computer systems, and software are multifunctional and sophisticated in their operation. As Sheil (1981a) points out, this complexity requires higher cognitive processing in order for a person to "cope." For less complex devices, like a toaster, a simple list of rules is adequate to explain its operation (e.g., put the bread in here then push down this knob). But for a computer, comprehension and understanding requires a person to build a conceptual representation (mental model) of its operation (deKleer & Brown, 1980; DuBoulay, O'Shea & Monk, 1981; Hoc, 1977).

The process of building a mental model requires several important cognitive skills. First, one must be able to organize many detailed, low level facts into coherent knowledge structure. Second, one must predict the operation of the computer on the basis of the model. This entails accessing one's knowledge representation in the act of writing a program. Third, one must compare the predicted with actual outcomes. Sometimes, such comparisons require interpretation of empirical evidence. Fourth, one must diagnose the cause of discrepancies between expected and actual outcomes. Lastly, one must refine one's knowledge so that the mental model is consistent with teh actual system. This process is iterative in nature, in that the cycle of prediction, testing, and refinement is repeated until an accurate model is constructed. Compared to other physical devices, the computer provides an advantage for the learner in that this adaptive process is safe, cheap, and fast (e.g., compare to costs of experimenting with nuclear reactors).

Sixth, computer environments for learning provide problems that have many correct solutions. Multiple solutions encourage divergent thinking, creative approaches to problem solving, individual expression and style, consideration of varied alternatives, and subjective judgment of appropriateness and elegance. In many current school learning environments, both the outcome and the process of achieving that outcome are fixed and evaluated by someone other than the student. Drill and practice software carries out this one method/one answer approach in computer environments for learning. However, the use of simulations, games, and other software offers the opportunity for students to spend some time in environments where more than one solution process can be successful and, in some cases, where the outcome of a task is not rigidly constrained.

These characteristics (interactiveness, precision, consistency, challenge, complexity, and provision for multiple solutions) are clearly found in certain traditional learning environments and therefore are not exclusive or unique to computer environments for learning. The point is that these characteristics are usually all present and present "at a high degree" in computer learning environments compared to other learning environments commonly found in schools and may lead to important differences in the acquisition of cognitive skills by students.

Pedagogical and Learning Theories Related to Software Design

An issue critical to the understanding of the cognitive demands and consequences of learning from software centers on the pedagogical and learning theories on which particular pieces of software are based. A concern recently has been raised that most software lacks any sound pedagogical basis, thereby decreasing the programs' potential educational effectiveness (Miller, 1983). This claim may have some validity, given that much of the software in the past has been designed by professionals outside the fields of education and psychology. However, it may be possible to identify the underlying theories of learning and instruction through systematic examination of the programs, despite the fact that such theories may not have been part of the original design specifications. It is important to note that the following discussion of the theoretical foundations of software is not based on any empirical evidence. Rather, it is an attempt to provide a general framework from which to examine software design and use. Furthermore, the discussion is not exhaustive, in that it presents only a limited number of theories that may be relevant to the argument. Additionally, the CAI/AICAI debate that is outlined briefly is only one of a number of issues that influences the implementation of software in educational settings. The debate, although important for software use, is more central to the controversies in software development. However, the theories and issues discussed provide one context from which to consider learning from software.

For example, much of the drill and practice and tutorial software reflects Thorndike's (1913; 1932) law of exercise which states that stimulus-response connections are strengthened by practice and repetition and are influenced by the effects of reward and punishment. Repetition with reinforcement is the basis of drill and practice and tutorial software. Additionally, these forms of software also reflect Gagne and Briggs' (1979) hierarchy of learning. Here, cognitive skills form a hierarchy, with higher-order skills built on lower-order ones. Tasks can be decomposed into the component skills and rules. Instruction subsequently can be structured from the bottom-up, starting with the most elementary skills on which other skills rest. Problems in learning can occur if certain lower-order skills within the hierarchy are deficient. A tutorial or drill and practice program can focus specifically on the deficiency, thereby allowing for remediation and further instructional progress. Such programs may also be seen in terms of a reception theory of learning in that the instructional materials are presented in final-form, leaving the learner simply to make a response. Similarly, these types of programs present the student with a stimulus-response learning paradigm. The computer generates the stimuli; the learner then must respond.

In contrast to the more didactic approach of drill and practice tutorials, educational games and simulations are for the most part, oriented toward a discovery learning paradigm. Students are forced, by the exploratory nature of the task to collect, analyze, transform, and integrate incoming data into existing cognitive structures. Learning by discovery actively engages students whereas learning by rule, in the form of tutorials or drill and practice, requires a more passive response from the students. Glaser (1966) characterized the discovery paradigm as sequences of trial and error learning from examples. Such a theory of learning is evident in games and simulations, whether in video games or educational settings. As Orbach (1977; 1979) and Bredemeier and Greenblat (1981) note, simulations provide an opportunity for realistic practice that would otherwise be impossible. Such learning from example is a critical part of discovery.

Additionally, many games and simulations are based on the Gestalt psychology concepts of productive and reproductive thought. These concepts translate cognitively into planning toward a solution, using a subgoal structure, and past experience to guide progress toward a solution. Reproductive thinking involves the direct application of previous learning to the present task whereas productive thinking entails restructuring past experience to fit the present situaiton (Wertheimer, 1959). It is likely that both productive and reproductive thinking are activated by games and simulations because of their balance between the hypothetical environment of the simulation and the real world. Past learning experience is necessary, yet it also may be engaged in a restructured form. Underlying these forms of thinking is the notion that a learner must manipulate the material actively in order to understand it. From this perspective, rote memorization is seen as an inefficient method of learning.

Yet, not all students are able to activate such thinking and

cognitive processes or profit from the autonomous, self-directed, and cognitively demanding discovery learning. Many learners require additional support from the instructional medium. Although "assistance" variations in CAI currently are being examined (see Burton & Brown, 1979; Goldstein, 1976; 1979; 1980; Mandinach, in progress), certain instructional and information processing variables have been identified as clearly influencing the theory underlying classes of educational software. Two of these variables are mentioned briefly here and discussed more fully in subsequent chapters. First is the cognitive load placed on students. Higher cognitive skills are more likely to be activated if tasks are cognitively demanding. Conversely, if little cognitive burden is placed on the learner, then the required cognitive skills will be correspondingly lower. Furthermore, learner aptitudes interact with cognitive load, indicating that some but not all students profit from the varying demands of software.

The second pedagogical variable that influences the structure of software is learner control. Proponents of student autonomy such as Merrill (1980) argue that students should be given the opportunity to structure their own learning environments by controlling the content and sequence of instructional materials. Conversely, opponents state that students are incapable of making informed decisions bout such important educational activities.

Issues such as learner control and the pedagogical theory of existing software form the basis of the present debate between proponents of frame-based CAI and AICAI (Miller, 1983). The frame-based CAI group charges that AICAI software lacks any theory of education or instruction. Students may or may not learn by discovery. Furthermore, the errors made in the learning by discovery paradigm are seen as senseless since the AICAI strategy is to islate "buggy" algorithms or cognitive processes only after they have occurred. The point being that an ounce of prevention is worth more than a pound of cure. Why allow the errors to occur at all and then remediate? Why not, instead, provide some instruction before allowing the learners to flounder in a trial and error manner?

Contrary to their opponents' charges, the AICAI paradigm does have an underlying theory of learning. Learning by discovery is not simply an instance in which students are left on their own to sink or swim. Rather, it is a theory of inductive learning and instruction that has important implications for long-term retention and transfer, intellectual self-regulation, and the acquisition of higher-order cognitive skills. Furthermore, proponents of AICAI see frame-based CAI as uninteresting in that the structure of the instruction eliminates many of the characteristics of software that stimulate motivation and cognitive engagement and fails to capitalize on the uniqueness of the computer environments for learning. However, there is a trade-off between the two positions of software development and their corresponding theories of learning. If instruction is made too explicit, thereby short-circuiting information processing and decreasing cognitive load, the resulting learning outcomes could be relatively small. Conversely, if there are excessive cognitive demands, the risk that the learners will fail to induce the necessary and intended information on their own will become increasingly apparent.

The purpose of the present discussion is not to argue for or against either theory of software development but rather to alert the reader to certain fundamental differences in the two perspectives. For the most part, the theories are not as distinct as is maintained by the CAI/AICAI debate. Software generally falls along a continuum from rote to discovery learning. The critical issue, however, is the way in which the programs are integrated into existing curricula and the pedagogical theories that guide their use. For example, a good tutorial can be used in creative ways, not only for remediation but also to stimulate higher-order cognitive skills. From the Gestalt perspective, if a student is required to do more than simply practice and memorize rules presented in the tutorial, the active manipulation of the concepts will facilitate and strengthen the cognitive outcomes. This is an instance whereby the corresponding instruction interacts with the underlying educational theory of the program, to influence and stimulate more high-level cognitive skills than would have been activated without addiitional instructional support. Similarly, games and simulations can be presented didactically, thereby eliminating the discovery learning component. Such a pedagogical shift will not only reflect the intent of instruction and the targeted outcomes but also effect the degree to which higher- and lower-order cognitive skills are activated by the software.

The point here is that the theories underlying the design of software are critical, but so to are the theories that guide their use in actual classroom or home learning environments. An example can be used to elucidate this notion. Recently it was brought to our attention that VisCalc, the electronic spread sheet by Visicorp, can be transformed into an educational game. A first reaction to that idea might be to ask why and how could such an expensive and complex business program be used in an educational setting. It seems clear that Visicorp's original intent for the spread sheet was very different. However, VisiCalc could be integrated into a number of existing curricula and taught from a variety of instructional perspectives as an educational game or simulation. It is important to keep in mind that educational software, like all forms of instruction, is only as good as the medium, content, and intent of the material presented. Learners will take from the instruction what they intend; no more and no less. The burden, therefore is placed on program designers, teachers, and researchers to make informed decisions about the selection of software appropriate to particular instructional purposes, and how best such software can be integrated into educationl environments.

CHAPTER TWO

Review of Relevant Research

Problem Situation

Individuals differ with respect to how they profit from instruction and learn most efficiently. Some students may benefit most from direct instruction with a teacher or from highly structrued instructional material, while others may require less instructional support or learn most effectively with computerized tutors. How to improve individual learning is an issue underlying the disciplines of cognitive, computer, and informational sciences as well as differential and instructional psychology. Students bring to an instructional situation certain psychological characteristics or aptitudes that influence how well they will learn under given conditions. Learners with different ability levels or aptitude profiles learn better under different instructional methods (Cronbach & Snow, 1977). Similarly, the media and symbol systems through which information is conveyed -- whether computer, textbook, or music -- affect comprehension of the material presented (Salomon, 1979). Such instruction also, through various demand characteristics, may stimulate certain cognitive processes in learners. In particular, the manner in which "assistance" variations in CAI may stimulate specific learner processes has only begun to be investigated (cf., Brown & Burton, 1978; Burton, 1981; Burton & Brown, 1979; Goldstein, 1979; 1980; Goldstein & Carr, 1977).

The issue addressed in the present review is that the demands of instructional tasks vary in explicitness. Whether classroom instruction or CAI, tasks focus on an obvious topic, but also convey in subtle ways knowledge of a more implicit nature, and may model or illustrate particular modes of cognitive behavior. All learning involves both implicit and explicit knowledge, yet individuals differ in the degree to which they actively process such knowledge, and the type of instruction presented can make different forms of knowledge more or less salient. For example, students may be given a reading passage on the construction of the Egyptian pyramids and told to remember important facts. This instruction explicitly cues the students to focus on significant dates such as an Egyptian dynasty or an historical figure like a certain pharaoh. Yet the students take more away from the learning situation than these details. They also learn more abstract lessons about the government, religion, and society of ancient Egypt. These concepts may not be directly discussed in the instruction, but the facts are discussed and the manner in which events unfold form an inherent structure on which the explicit facts are based. Some students will comprehend only the significant details from such a reading, while others will go beyond the presented information to plan, organize, and understand the implicit material. If the instruction is

unstable, as is traditional classroom teaching, even more peripheral information (e.g., teachers' attitudes) may be inferred; stable instruction such as CAI can control excessive inferencing. In either case, strategic planning knowledge is of critical importance.

Greeno (1978) defines strategic planning knowledge as the ability to set goals, choose appropriate action plans, and in general, organize cognitive activity on a problem that will produce a solution. Strategic knowledge, as Greeno points out, is generally not an explicit part of the curriulcum, although "it seems likely that many students acquire strategic knowledge by induction from example problems that present strategic principles, implicity" (p. 72). Furthermore, there appears to be two types of knowledge (Greeno, 1972) and two types of cognitive structure variables (Mayer & Greeno, 1972) that interact and influence acquisition of strategic planning knowledge. Greeno (1972) distinguishes between propositional knowledge (which relates to properties of and relations among objects, and conceptual information), and algorithmic knowledge (which deals with procedures, rules, and operations).

The structural variables concern the degree to which instructional and cognitive components are connected and relate to the type of knowledge required by a task. Internal connectedness refers to the relations among the components within cognitive structure, while external connectedness refers to the degree to which the components are related to others in the cognitive structure (Mayer & Greeno, 1972). Studies have tested the interactions among knowledge and structural variables (Egan & Greeno, 1973; Mayer, 1974; 1975a; 1975b; Mayer & Greeno, 1972). Different methods of instruction seemingly result in qualitatively and structurally different learning outcomes. Support for the interactions is evident in the type of tasks to which instructional methods transfer. The type of knowledge and the connectedness of components influence the ability to set goals and organize cognitive activity.

A logical extension of the knowledge-structure interaction is to examine the relations among aptitude, knowledge, and cognitive structure. Snow (1980a) speculates about possible interactions between the explicitness of instruction, connectedness, and aptitude profiles.

> Able learners who can elaborate their own structure and discover the necessary strategic knowledge by induction will show well-intergrated cognitive structure and facile transfer to novel problems. Less able learners might have good internal connectedness for pattern recognition and propositions explicitly taught, but they are less likely to have discovered the implicit strategic principles or elaborated transferable external connections. Auxiliary instruction designed to provide the strategic knowledge directly would be expected to reduce the aptitude-achievement relation by helping those less able students who would not induce strategic knowledge or elaborate external connectedness on their own; direct instruction in strategic knowledge could produce active rules for them to follow but it might reduce transfer. If the act of discovering and elaborating strategic knowledge on their own produces

> good external connectedness and transfer for able students, direct instruction in this might well produce a decrement in their performance (p. 53).

The present review examines the literature as it relates to a series of proposed studies that focus on the degree to which ability and other individual differences interact with the explicitness of cued instruction on computerized problem solving tasks. Students who learn only details may or may not benefit from the cued instruction in strategic planning. Processing cues may need to be particularly salient and controlled for such students; this is possible in CAI. Other students can induce information about planning that is implicit in less salient examples. Such differences across students should be predictable from their scores on standardized achievement tests (Snow, 1980a; 1982).

Definition of the Research Problem

A multidirectional approach to the investigation of the cognitive effects of computer assisted learning is necessary because much of the existing literature fails to focus specifically on the cognitive demands and consequences of computer environments for learning. However, a number of literatures either address the issue to some degree or the findings are seen as relevant to a better understanding of computer assisted learning. Consequently, a comprehensive review must consider the convergent perspectives of such literatures as (a) individual differneces and ATI (aptitude-treatment interaction) research, including aptitude and ability difference as well as cognitive styles and affective factors; (b) problem solving and knowledge states; (c) instructional psychology; (d) planning; and (e) CAI, computer tutor and artificial intelligence. While no single area of research can provide such a comprehensive understanding, the present review attempts to integrate five somewhat disparate literatures into a cohesive perspective of the cognitive demands and consequences of computer assisted learning.

Individual Differences

It seems apparent that the cognitive demands of computer learning are complex and involve higher-order and lower-order cognitive processes as well as more general cognitive styles. These variables, in turn, interact with enduring psychological learner characteristics such as aptitudes and ability profiles. Aptitudes interact with the type and medium of instructional treatment. No one treatment is right for all students, nor is a particular instructional method compatible with all media. The literature on individual differences in aptitudes, abilities, and cognitive styles has particular relevance for the present review.

Aptitude and Ability Differences and ATI Research

Ability differences among learners were distinguished as early as 1512 when Nicolo Machiavelli wrote The Prince. At the time, Machiavelli provided advice as to the type of ability a Renaissance prince should

seek in his ministers. The philosopher wrote:

> There are three kinds of intelligence: one kind understands things for itself, the other appreciates what others can understand, the third understands neither for itself nor through others. This first kind is excellent, the second good, the third kind useless (p. 124).

The distinction remains relevant today and can be extended to current educational practice. Some students can learn with minimal instructional support while others require more detailed and intensive instruction to understand or perform efficiently.

Translated into current psychological terms, Machiavelli's first kind of intelligence is characteristic of self-sufficient or self-regulated learners (Corno & Mandinach, 1983). These students accept responsibility for their learning and actively confront the information processing demands of the task. The second type of intelligence characterizes learners who can perform effectively by availing themselves of what is in the task environment. Corno and Mandinach call this kind of learning resource management. The inability to distinguish between relevant and irrelevant information on one's own can be accomodated by getting outside assistance, or managing resources. Machiavelli failed to distinguish a similarly weaker kind of learning, which can be called task focus. Here the student discerns relevant from irrelevant task information, but fails to go beyond information given when the task may require it. Learners who exhibit both these weaker forms of intelligence can benefit from instruction that balances the placement on and removal of information processing burdens from the learner. The final type of intelligence poses a greater challenge for instructional psychologists. Such students learn by intensive and explicit instructional support, and low processing burdens. Allowing the environment to assume responsibility for one's learning, rather than taking an active role in the process, may be termed recipience. Whether political ministers in Machiavelli's era or present day students, individuals do differ in intelligence or general mental abilities and therefore differ in the extent to which they benefit from various forms of instruction.

Definition of aptitudes and ATI. Individuals bring to an instructional situation characteristics or aptitudes that influence their performance on a given task. Snow (1980a) defines such aptitudes as the "psychological characteristics of individuals that predispose and thus predict differences in later learning under specified instruction conditions" (p. 41). It is not enough simply to examine the effects of aptitude differences on performance. Nor is it sufficient to study performance differences across instructional methods in isolation. Snow (1976b) provides three recommendations for future ATI research:

1. The examination of the most plausible ATI hypotheses in large-scale, long duration, real school studies.

2. The development of methodology capable of handling the complexities of such research.

3. The development of a laboratory science for the analysis of aptitude tests and learning tasks, and the ATI constructs based on them (p. 51).

Critical to ATI research is the definition of aptitude. Cattell (cf. 1963; 1971) divided general mental ability (g) into crystallized (gc) and fluid (gf) ability. Crystallized ability refers to previously constructed assemblies of performance processes retrieved as a system and applied anew in familiar instructional situations (Snow, 1980c; 1982). Crystallized ability often is seen as general achievement since the construct reflects the long-term accumulation and organization of knowledge and skills. Fluid ability refers to new assemblies or flexible reassemblies of performance processes to adapt to new and unfamiliar situations and novel problems. These abilities form a complex of cognitive structures responsible for the assembly, reassembly, acquisition, and organization of knowledge and skills into functional performance processes.

Aptitude profiles interact with instructional methods to produce differential learning outcomes. The goal of ATI research therefore is to "understand, considering all relevant personal characteristics, what makes a person ready to profit from a given kind of instruction, why persons differ in this readiness, and how instruction can be designed to fit each person's readiness in order to promote optimal achievement, under specified conditions" (Snow, 1980a, p. 42).

Apparently, one of the few consistent findings that speaks to this objective concerns the extent to which information procesing burdens are placed on or removed from the learner. Snow (1976b) reviews the literature and outlines 20 instructional treatments that can be classified according to processing demands. Treatments that relieve information processing burdens benefit low ability learners and apparently interfere with pre-existing knowledge structures of high ability students. Conversely, treatments that place the processing demands on the learner profit the high rather than the low ability students. Gray (1982) examined the relation between cognitive organization and aptitude. She found that high ability students organized and imposed structure on the instructional material while the low ability students exhibited little organization, elaboration, or integration skills. Yalow (1980) found similar results using instructional material from different content areas. Instructional material that relieved information processing burdens through simplificaiton, structuring, or elaboration aided the low ability students, while high ability learners profited from minimal treatment. Yalow also found that instruction that provided short-term benefits did not enhance long-term learning. Conversely, instruction that aided long-term retention did not benefit immediate performance. In other studies, high ability students benefited from instruction concerning models and concepts as opposed to rote algorithms (Mayer & Greeno, 1972), and discovery learning as opposed to learning by rule (Egan & Greeno, 1973). In general, patterns of results appear to show that highly supportive instruction benefits the low but not the high ability students since the treatment may interfere with rather than facilitate the latter students learning. High ability students tend to perform better when given less instructional support because they can impose previously successful procedures on the learning task (Tobias, 1982).

Instructional treatments that relieve processing burdens require less cognitive work from learners. Salomon (1979) has characterized two burden-reducing instructional strategies as short-circuiting and supplanting (or modeling), respectively. In short-circuiting, the instruction is organized and carries out the processing for learners who do not do it on their own. In modeling, the instruction uses films or video examples from which students observe and learn the presented material. Ausburn and Ausburn (1978) futher differentiate two types of supplantation treatments that form the basis of an instructional design model, while taking into consideration students' cognitive styles as they interact with instructional tasks. Conciliatory supplantation alters the mode of task presentation such that the difficult elements are removed, leaving the student with a prefered instructional model. Compensatory supplantation performs what learners cannot do for themselves.

Salomon (1974) also distinguishes between preferential and compensatory treatments. The former focuses on tasks that require mental operations available to and favored by a student. The latter provides skills students lack to complete the task. These treatments also can be seen as varying in the level of instructional explicitness. Here explicitness ranges from a compensatory treatment that provides maximal instructional guidance to activating treatments that provide little or no guidance, thereby requiring students to use their own strategies. Results indicate that low ability students benefited from the highly explicit or compensatory treatment while high ability learners profited from the activating or preferential treatment. However, Carrier and Clark (1978) found conflicting results. Both high and low ability students profited from explicit instruction. Research on the effects of explicitness, persistent mode, and processing burden, as they relate to differential human abilities vary across selected tasks, operationalization of ability level and instructional treatment. It does seem clear that learners with different ability profiles process instructional information in diverse manners.

Self-regulated learning and meta-cognition. Corno and Mandinach (1983) outline a model for how students process instructional information. The model revolves around self-regulatory processes -- including alertness, selectivity, connecting, planning, and monitoring. These processes are seen to function as either information acquisition or reduction processes. Alerting and monitoring are critical for acquisition (and are generally higher-order). These, in turn, control the lower-order reduction processes -- selectivity, connection of new to old information, and the task specific planning of performance routines. Students using both types of processes are seen to be selfregulated; they engage on their own the necessary cognitive activity for processing information. Conversely, when students do not actively process instructional information, they are seen to be recipient learners. This kind of learning allows the instructional environment to perform most of the needed cognitive activity. The student can respond passively to the learning environment. Recipient learning should be most fostered in Salomon's short-circuiting and Snow's low processing burden instruction.

In a third type of learning, the student exhibits a style of information processing called resource management. The student uses information acquisition processes but avoids the extra mental work of information by getting this assistance externally. Some types of cooperative learning (Slavin, 1980) may promote resource management. Finally, there is the learning strategy that is regularly higher in information reduction than in acquisition, termed task focus. Here the learner selects and connects new information to existing knowledge, and engages in planning, but is less inclined to gather a wide variety of information, to elaborate information, and to monitor performance carefully. Such an approach should be fostered under conditions of individualization that also entail problem solving and logical reasoning but do not require continuous monitoring or elaboration.

Students who characteristically adopt one or another learning strategy seem likely to benefit from different types of instruction, depending on the task objectives. The distinction between recipient and self-regulated learning, in particular, seems to characterize the observed strategic differences of low and high ability learners, as measured by general ability tests (Gray 1982). Low ability students should profit from instruction that emphasizes self-regulated learning but also relieves some of the processing burden. Guided modeling instruction fills this order. It allows students to observe self-regulated learning skills while not requiring full processing until after having practiced such skills with assistance, then on their own. Since high ability students appear to engage spontaneously in self-regulated learning (or at least task focus), even when short-circuits are provided, they should benefit from instruction that activates (Salomon, 1979) their own processes and thereby challenges their learning. Perhaps an attempt to stimulate self-regulated learning processes in both high and low ability leaners, using a CAI situation that might, without appropriate instructional cues, promote resource management in low ability students and task focus in highs. Learners of different ability levels who are provided instructional support of varying intensity should differ in the extent to which they induce self-regulated learning processes on their own or through acquisition from the instructional material.

Brown and DeLoache (1978) and Flavell (1976) approach the issue of self-regulated learning from a cognitive developmental perspective. Flavell uses the term metacognition while Brown and DeLoache refer to self-interrogation and self-regulation. Brown and DeLoache note that novices and children may fail due to a lack of ability, but also because they do not regulate and have a conscious awarenesss of their actions. If a child is insufficinetly familiar with a given task then that lack of knowledge will be manifested in inadequate monitoring and activity selection toward the ultimate goal. Such familiarity is experientially based. With time, children can bring to bear such meta-cognitive skills as predicting, monitoring, testing, and controlling one's actions toward a goal. Hayes (1976) reports that college students can be taught to self-monitor and that the effects are evident in increased self-awareness, improved diagnostic and organizational planning and problem solving skills.

From a different perspective, Goldstein and Papert (1977) and Teske and Pea (1981) stress the role of self-knowledge in artificial intelligence. Self-observation, allows a program or a learner to monitor and regulate performance, learn by debugging errors, and generalize knowledge to other tasks and situations.

Information processing components of individual differences. A rather different perspective on aptitude and ability differences is provided by Sternberg's (1982) componential approach to intellectual development. An elementary information process, called the component operates on internal representations of stimuli. Sternberg distinguishes between five kinds of components: metacomponents, or higher-order control processes; performance, acquistion, retention, and transfer components. Individual and developmental differences are apparent across these components. Sternberg (1980) proposes six meta-components that develop with age and and on which there are differences across individuals. First, students must identify that a problem exists and determine what is needed to solve that problem. Problem recognition, as Resnick and Glaser (1976) point out, can be considered one definition of intelligence, in that it is the ability to function effectively without direct, explicit, or complete instruction. Second, students must be able to select the lower-order components needed, given the task and situational demands. They then must select an appropriate strategy for combining these lower order components and a representation for organizing the information. Fifth, students must decide what and how many resources to apply to the task at hand. A major issue concerns allocating encoding time. Apparently, successful learners spend more time initially encoding the stimuli then do less effective students (Sternberg, 1977; 1983). The premise here is that an initial investment to understand the problem subsequently leads to a more efficient use of learning time. A related issue concerns the use of global versus local planning and solution monitoring. Sternberg (1981) reports that better reasoners use global planninng techniques whereas poor reasoners use more local planning. Local planning is a microstrategy to be applied on a specific item while global planning is a macrostrategy that is applicable across specific items.

The difference may be seen as analogous to what Miller, Galanter, and Pribram (1960) refer to as tactics and strategies. Strategies are considered molar units of analysis while tactics are more molecular in nature. Johnson (1971; 1978) also distinguishes between tactical and strategic methods. Strategy is equated with preplanning activity whereas tactical involves little preplanning. The learner chooses a course of action following the outcome of the preceding trial. These distinctions reflect the differences between reactive (tactical) and proactive (strategic) planning behavior. Preliminary results from a study of strategic planning knowledge using an instructional computer game lend support to this notion (Mandinach, in progress). High ability students spent more time encoding the stimuli and problem situations and planned globally at critical decision points. Latency patterns indicate that when a decision point is encountered, successful students pause to gain an understanding of the problem space, plan several moves ahead, and then make a response to the keyboard. Conversely, low ability students spent less time on encoding and what planning they performed tended to be locally relevant. These students did not plan ahead but rather made local decisions at each computer prompt.

Furthermore, Sternberg (1982) notes certain external variables that influence the development, acquisition, and manifestation of various components. The list includes (a) multiple occurrences of the target information; (b) variability in contexts for presentation of target information; (c) relevance or importance of target information to the overall situation; (d) helpfulness of surrounding context to the understanding of target information; (e) location of helpful information with respect to the target; and (f) helpfulness of stored information to understanding of target information. If these variables, as Sternberg suggests, influence the acquisition of knowledge, it seems apparent that each is amenable to the computer, as medium of instruction. Not only can these variables be designed into software but perhaps their influences can be enhanced and studied more effectively, given the unique characteristics of a computer learning environment.

Formal reasoning. Another ability factor that may be associated with response to computer learning environments has been referred to as formal reasoning by Inhelder and Piaget (1958). The Adolescent Reasoning Project at the Lawrence Hall of Science has conducted extensive research on formal reasoning and examined the correlates between formal reasoning and a number of other problem-solving areas (Linn, 1978). It seems clear that measures of formal reasoning predict performance in complex problemsolving situations such as learning to program computers. It is somewhat less clear what these tasks measure. In particular Clautier and Goldschmid (1976) found very little variance in formal reasoning measures when they examined measures of fluid and crystallized ability in a regression equation. Similar results have been found by Linn, Pulos, and Gans (1981). Humphreys and Parsons (1979) have suggested that Piagetian reasoning is not different from other measures of general ability. This work involved students at the concrete operational stage, however, not those at the formal stage. Nevertheless, there is some controversy concerning the distinctness of formal reasoning from other measures of general ability.

A myriad of tasks have been developed to measure formal reasoning actually and are only slightly related to each other. Martorano (1977) for example, found almost no overlap between various measures of formal reasoning in her study. Pulos and Linn (1981) found a communality of only about 11 percent of the variance when they used formal reasoning tests representing different contexts but the same underlying formal reasoning strategy. The Adolescent Reasoning Project has a group administered task called Predicting Displaced Volume which not only specifically measures formal reasoning but also categorizes the respones of individuals into four different possible strategies. Thus, one not only can conclude level of the student's reasoning but also the actual processes that the student is using to solve the problem. In particular, certain erroneous responses to the Predicting Displaced Volume problem apparently reflect inability to consider all the alternatives in the situation. Students fail to consider that the weight of an object, which is important in many situations, may not be important in all situations.

Cognitive styles and affective factors.

The demands of computer learning not only involve stable, psychological characteristics such as aptitude and higher-order cognitive

skills like strategic planning knowledge, but also engage psychological constructs that may be termed cognitive styles. There are certain cognitive styles that are particularly applicable to a computer environment. However, first it is necessary to distinguish between cognitive styles, strategies, and abilities before discussing a number of cognitive styles and their relationship to computer environments for learning.

Cognitive styles, strategies, and abilities. Perhaps the most comprehensive work on cognitive styles and their implications for educational practice is provided by Messick (1982; 1983). Cognitive styles are characterizzed as stable and pervasive information processing regularities that develop around personality styles. Furthermore, cognitive styles are not easily trainable nor amenable to change. They are spontaneously applied and seen as dimensions of continuous variations. Messick (1982) describes cognitive styles as "consistent individual differences in ways of organizing and processing information and experience" (p. 3). Conversely, cognitive strategies are seen as conscious or unconscious decisions about alternative behavior, determined by task requirements, problem content, and situational constraints. Strategies are more amenable to change via training, such that more effective strategies can be called upon, given specific task requirements.

Messick (1982; 1983) also distinguishes between cognitive styles and cognitive abilities. Research on cognitive abilities focuses on the content and level of cognition asking questions about "What?" and "How much?". The emphasis is on competencies and maximal performance. Conversely, research on cognitive styles asks "How?" and focuses on propensities and typical performance. Cognitive styles are seen as bipolar, with each of the extremes having different implications for cognitive functioning. Messick terms such functions as value differential in that each extreme is adaptive for different circumstances. Cognitive abilities are unipolar and value directional in that more rather than less of an ability always is considered desirable. Furthermore, abilities are seen as domain specific while styles cut across domains. Finally, styles are considered high-level heuristics for organizing and controlling variables whereas abilities are enabling variables that facilitate task performance.

Field dependence/independence. Messick's distinctions are critical to an understanding of the cognitive processing involved not only in computer assisted learning but learning, in general. A number of cognitive styles have been hypothesized to relate to computer environments for learning or to the likelihood of acquiring higher order cognitive skills. These stylistic variables include field dependence/independence, a construct identified by Witkin (1949) to measure the likelihood of focusing on the situation in the task (field independence) as opposed to the likelihood of considering contextual information not necessarily central to the task (field dependence). Such a style characteristic may influence response to computer assisted learning. Field dependence/independence relates directly to the precision of computer environments for learning in that precision involves the generation of alternative problem solutions and the presentation of task focused information. Instructional material presented on a microcomputer can minimize the amount of extraneous information a student receives. Effective learning involves the ability

to distinguish between relevant and irrelevant information. Computer assisted learning also exercises students' ability to deal with ambiguity (Corno & Mandinach, 1983). They first must be able to identify that ambiguity exists and then determine how to deal with such inconsistencies. Field dependence may be manifested in a computer learning environment when students attend to irrelevant rather than relevant information. If students are unable to select the appropriate data for particular problems, then problem solutions could be less efficient. In contrast, students may manifest field dependence by attending to details and selecting information appropriate to the particular problem.

Reflective thinking. One cognitive style variable that seems relevant to learning in computer environments is "reflective thinking" as proposed by Dewey (1933) rather than the more recent cognitive tempo style known as reflectivity/ impulsivity (Kagan, 1966). Reflective thinking, in Dewey's terms, refers to the considerations of alternatives and reasons prior to attempting a solution or choosing a course of action. Baron (1981) elaborating on Dewey's model, presents five components of reflective thinking (problem recognition, enumeration of possibilities, reasoning, revisions, and evaluation) that are applied differently by individuals and are seen as propensities rather than abilities. Each of Baron's five phases of reflective thought seem particularly relevant to both programming and computer learning environments. Differential values on these organizing and controlling variables, Messick's terms (1982), should influence the level of cognitive functioning and the skills acquired through instruction in computer environments for learning.

Risk taking. The willingness to take risks is a third cognitive style variable seen as critical to effective computer learning. If students are unwilling to take risks and attempt novel solutions, then their knowledge and repetoires of effective solutions will be limited severely by the constraints of their risk-taking behavior. Errors stimulate learning by emhasizing the inappropriateness of particular algorithms applied in certain situations. Students gain knowledge of those errors through both correct and incorrect applications. The continuous and precise feedback provided by the computer helps the student to learn rapidly about particular bugs. The more risks students are willing to take, the more and deeper knowledge they will acquire about effective learning.

Computer-related anxiety. The precision of the environment involved in computer learning also may lead to anxiety on the part of those individuals who dislike being wrong and have difficulty coping with that. This has been referred by some as fear of machines, which seems perhaps a bit overgeneralized since very few individuals are afraid of machines such as refrigerators. However, more specifically, and from a process standpoint, it would seem that a stylistic variable based on anxiety related to possibility of failure in a precise environment, and therefore to the likelihood of receiving negative feedback might be important in response to instruction in a computer environment.

Additionally, anxiety may be manifested simply through interactions with the computer. Many students believe that microcomputers are mysterious and complex blackboxes that will break if abused. The reactiveness

and precision of the computer environment may be another cause of some student's computer anxiety. Once such initial anxiety is overcome, computers become challenging and educational environments in which to test novel problems that might not be attempted in a traditional classroom. The computer becomes an interactive and respondent medium that reacts to the student, and the student to the computer.

Attributions and expectations. Another factor, frequently associated with response to precise environments, concerns attribution of failure. Performance attributions, and in particular, attributions of failure, are tied closely to computer anxiety and the unwillingness to take risks. Attributions of failure to internal causes, such as a lack of ability, can harm not only a student's performance but also the psychological approach to a given task (cf. Diener & Dweck, 1978; Dweck, 1975; Dweck, et al., 1978; Dweck & Gilliard, 1975; Dweck & Reppucci, 1973; Ruble, Parsons, & Ross, 1976). In computer learning, students come to learn that failure is not to be avoided, but rather that it can provide valuable feedback generalizable to other situations. Failure is also a relative phenomenon. In computer learning, it should be more an educational than a negative experience, since students not only learn from mistakes but also gain critical understanding from remediation of the errors. It is possible that attributional patterns in computer assisted learning differ from those observed in more traditional media because of the unique characteristics of computer environments for learning (Mandinach, in progress). Preliminary results indicate that especially for girls, failure is attributed to mechanical characteristics of the computer and success to luck. Boys, however, attribute both success and failure to strategic approaches to the task.

In a related manner, students who tend to avoid and are fearful of evaluative feedback may show the negative effects of computer learning as they do with other instructional media (cf., Brickman, et al., 1976; Dweck, 1976; Dweck & Bush, 1976; Nicholls, 1975; Parsons, et al., 1976). The precision of the computer environment and the consistency of feedback should influence a student's level of anxiety toward the educational medium. Anxiety concerning evaluative feedback many times is related to inconsistency and unfulfilled expectations. Once the student comes to realize that the feedback is constructive and consistent, rather than unstable, as is the case in much traditional classroom instruction, then the student can begin to build stable performance expectations, which, in turn, should have a positive effect on learning and anxiety. Students should be able to formulate stable performance expectations while gaining consistent experience with the computer.

The relation among cognitive styles, aptitude, higher-cognitive processes, and computer environments for learning remain unexplored and therefore are only hypothesized. The literature examined here relates to instructional domains other than the computer. But as Messick (1982; 1983) points out, cognitive styles are pervasive and cut across content domains. It therefore seems possible that the cognitive styles that are critical to the acquisition of knowledge in other content areas and media of instruction also should be important to computer assisted learning.

Instructional Psychology

Learning by rule, discovery, induction, from examples, or other instructional methods all require the activation of cognitive processes that are specific to the form, content, and medium of instruction. The support, cognitive demands, and characteristics that underlie an instructional method influence how, why, and to what extent students profit from specific treatments. Tobias (1982) defines instructional support as "assistance given the learner by way of organizing the instructional content, maintaining student attention, eliciting responses, providing feedback on the responses, and so on" (p. 5). Glaser (1982) presents a more global perspective from which to examine the underlying facets of various forms of instruction. Glaser outlines four components that characterize instructional psychology: (a) an analysis of the competence to be attained and the structure and processes of the desired knowledge state, (b) specification of the initial state of the learner, (c) specification of the conditions for learning and the transitional processes required to decrease the distance between the learner's initial and desired knowledge state, and (d) assessment of the new knowledge state. These components not only account for variation across instructional treatments, but also consider individual differences among learners and their initial states of performance. These are critical factors in the design of ATI research. The instructional psychology literature is far too vast to examine in depth here. For the purposes of the present review, research that focuses on task transfer, learning by rule, induction, discovery, and learning from examples will be reviewed.

Transfer

One possible objective of instruction is transfer, yet there is little evidence to support the hypothesis that one cognitive task affects performance on another. As Bruner (1973) notes the essential question here is under what constraints will students learn so as to optimize the chances of generalizing that knowledge to other situations. Royer (1979) presents a comprehensive discussion of the current state of research and theory on the transfer of learning. Several theories and their distinguishing characteristics can be classified into two categories of transfer. The first class of theories concerns learning and transfer across tasks that share common stimulus properties. Royer points out that these environmental theories state when transfer should occur, thereby aiding in the sequencing of instructional activities, but fail to relate school to real-world learning problems. In contrast, theories based on the likelihood of retrieving relevant knowledge and skills (cognitive theories) lack direct educational applicability but provide a link between learning in academic and real-world settings.

Environmental perspectives, such as the theory of identical elements, assume that transfer is limited to similar learning situations in which students recognize the commonalities among tasks. Skills or knowledge learned on one task should transfer to other learning activities that require similar skills. A general problem with environmental theories is that the common elements across tasks may not be obvious to

the learner, thereby preventing transfer. Another problem is that environmental theories are based on stimulus-response connections and ignore cognitive activity. It is difficult to isolate identical elements if task demands and required cognitive processes have not been identified and examined. Alternatively, the cognitive theories rest on the assumption that the probability of retrieving relevant knowledge infleunces the extent to which task transfer occurs. Transfer then becomes a matter of improving the connections among cognitive structures so that the appropriate information is accessible upon demand. Transfer can be viewed from the perspective of schema theory in that it occurs when the cognitive demands of a new task activate pre-existing schema from memory. One problem with cognitive perspectives such as schema theory is that they presently lack clear definitions for the purposes of educational applications.

Additionally, Royer has outlined two global problems in educational practice that concern transfer. First, how can instruction be arranged so that previous knowledge and performance can facilitate current learning? The second problem concerns the arrangement of instruction, so that learning in academic settings can be applied in real-world situations. These problems seem like logical educational objectives, but obtaining such transfer is no easy task. Evidence of successful performance on specified tasks is hard enough to obtain. Inclusion of transfer measures makes the research endeavor all the more difficult. Yet transfer is used as a criterion by which to judge the success of instruction and therefore is included as an outcome measure in many problem solving and instructional studies.

Rule and Discovery Learning

One approach to the use of transfer tasks in instructional research is reflected in Egan and Greeno's (1973) work on rule and discovery learning. Students received either learning by rule or learning by discovery instruction; then were tested on transfer problems. Apparently, the two instructional methods led to different cognitive structures and knowledge states as reflected in transfer performance. Learning by rule was related to near transfer (students performed well on similar tasks), while learning by discovery reflected far transfer (students performed well on less similar tasks). The authors conclude that "the outcome of discovery is the structural integration of previously known concepts, while the outcome of rule learning is the addition of new structure" (p. 85). Discovery learning involves more problem solving, generalization, long-term retention, and transfer. Rule learning on the other hand, is superior for immediate tasks. Learning by discovery influences the integration of new information into existing cognitive structures and allows for direct applications of the acquired knowledge to other tasks. In contrast, learning by rule does not stimulate reorganization of the cognitive structure but instead simply causes addition of new to the old knowldge stated without integration. Mayer (1974) and Mayer and Greeno (1972) found similar patterns of transfer with instructional methods that emphasized operations or concepts and concluded, based on the results of the transfer measures, that the two groups differed structurally and qualitatively on learning outcomes.

Much attention has been devoted to examining the differences between discovery and rule learning (e.g., Shulman & Keislar, 1966). Wittrock (1966) makes the distinction that learning by rule is a passive method, whereas learning by discovery is an active one. Kagan (1966) characterizes discovery learning as an arousing and motivating method that requires active involvement on the part of the learner. Cronbach (1966) states that it instills in the learner a sense of "intellectual self-reliance." Another distinction is that discovery and rule learning are analogous to induction and deduction (McKeachie, 1974; Wittrock, 1966). Discovery learning requires moving from specific to general concepts; rule learning provides the student all the information needed to solve the problem.

Learning from Examples and Induction

Glaser (1966) has characterized discovery as inductive sequences of trial and error learning from examples. Students are given examples that should stimulate induction of general propositions. A trade-off between the risk of errors and the explicitness of the rules is critical here. When examples are presented before the rules there is a risk of inducing incorrect rules, whereas the satisfaction of discovery is minimized when rules precede the examples. Rule-example sequences decrease information processing burdens and are instruction-driven. Example-rule sequences are student-driven, allow learners to impose their own cognitive structures on the instructional material and are more oriented to discovery learning. Yet there is always the risk that the students will fail to induce the necessary information from the example-rule sequences, or fail to engage in self-regulation. However, when Anderson (1967) summarized the effects of rule-example and example-rule sequences, he found that the former tends to produce faster learning, fewer errors, and better retention. The appropriateness of this generalization across students with different aptitude profiles and cognitive styles has yet to be confirmed.

Results from an examination of human inferential processing by Collins, Warnock, Aiello, and Miller (1975) indicate that learners are capable of extracting "what they do not know explicity from some forms of explicit knowledge by plausible but uncertain references" (p. 414). When faced with incomplete knowledge, students can all upon general reasoning and heuristic strategies in an attempt to complete that knowledge structure. Additionally, Hayes-Roth and McDermott (1976) found that students can induce knowledge by abstracting representations of structured patterns from sequences of examples. Students must identify common properties among examplars, using processes called matching and interference. Students learn by comparison, recognition of novel instances of general rules, discrimination of relevant from irrelevant properties thought to be critical, and from practice of rule applications provided in instructional examples (Hayes-Roth, 1978). These are all self-regulated processes. Sequences of examples that contain both relevant and varying irrelevant factors are administered, thereby requiring the students to identify the critical properties and induce relations among the examples. Similarly, Ali (1981) points out that it is important to include both positive and negative examples so that students

can focus on relevant facets of the stimuli, discriminate against the irrelevant, and learn to distinguish when the rule is or is not applicable across situations. The student is not permitted to use resource management or task focus in these studies.

From a slightly different perspective, von Oech (1982) discusses the generation of creative ideas and problems solving skills and how positive and negative examples can stimulate the thought. Failures are seen as stepping stones to new ideas rather than instruments for negative feedback. Errors are indicators not only of the need to change direction but also that the problems students may be attempting divergent approaches to the tasks that are beyond their level of experiences. Such errors should not be discouraged since they may stimulate new and creative approaches to the specific task and learning, in general.

Cohen and Feigenbaum (1982) consider learning from examples to be an inductive process. The authors distinguish three types of tasks that vary in complexity. First and simplest is the classification or prediction tasks with only one rule or concept. Second, there are tasks with multiple concepts. Finally, there are planning tasks that require execution of a sequence of rules. Learners must integrate new rules and diagnose incorrect rules as tasks increase in complexity. They also must generalize from the given examples rules that are applicable to the problems. This requires a transformation to higher-order knowledge so that the examples can be used effectively. Such generalizations from examples to a broader class of tasks is the critical inductive inference process.

Several theories exist that examine the rules (Polya, 1954), structures (Hunt, et al., 1966), and relations to problem solving (Egan & Greeno, 1974; Simon & Lea, 1974) in learning from induction literature. Polya (1954) views learning from induction in mathematics as a series of observations, conjectures, generalizations, and tests of reasoning. Induction begins with observation, during which students note similarities among examples and form conjectures based on available evidence. The students must be willing to revise any conjectures when new evidence supports the need for change, but not make any changes without good reason. Finally, the student tests the conjecture against the new evidence and starts the inductive process anew if lacking confirmatory evidence. Continuous monitoring is critical to the conjecture-test cycle in order to obtain verification or disprove a particular case. Similarly, Hunt, Martin, and Stone (1966) developed a concept learning system in which concepts are seen as a sequence of tests and decisions based on inductive reasoning. Sequences take the form of decision trees with interconnecting nodes that represent the flow of reasoning decisions throughout the network. Each node reflects a decision point at which the student must induce, from the available information, a path through the unfolding decision tree. Again, continuous testing and monitoring of decisions are critical to learning or obtaining correct solutions.

Egan and Greeno (1974) also view instruction as a network of nodes that reflects the presence or absence of specific features. Tests at each node depend on the sequence of previous decisions and outcomes. Learners select relevant properties from the information given and

attempt to discover the correct patterns of branches by connecting appropriate nodes within the network. Through a process of hypothesis selection, assessment of responses, resampling after errors, and coverging on possible solutions, students then are able to abstract the rules and structure of the network. Each test of a conjecture provides students with additional information from which to formulate the next hypotheses. The iterative nature of induction is critical here and is made simpler or harder based on the explicitness of the knowledge in the problem space. If the problem space is explicit, the iterations become a sequence of tests and hypothesis elimination. Successful performance depends on how learners process the untested rules and account for the previously rejected rules, i.e., engage in self-regulation for knowledge given. In contrast, when the problem space requires implicit knowledge, successful performance rests on learners' ability to observe and integrate the relevant features in an appropriate manner. This imposes a heavier processing burden on learners because accepted and rejected hypotheses must be stored and retrieved upon demand as the students continue to scan, abstract, and assess the relations among the current set of features. They must engage in self-regulation with knowledge external to the task as well.

Pellegrino and Glaser (1982) state that the keys to inductive learning are better understanding of the task constraints, induction of a knowledge structure by detecting relational patterns, and discovery of connections between new and old information. The authors identify six processes that underlie inductive reasoning tasks. First, encoding and representational processes concern the storage and retrieval of relevant information. Inference processes relate common elements among elements in memory. Complex knowledge structures are transformed into simpler ones through rule assembly and monitoring processes. Evaluation of similarities among elements is performed through comparison processes, followed by discrimination processes that select relational structures, and finally decision processes that generate responses to the learning situation. Pellegrino and Glaser have fround that these processes and performance on inductive tasks correlate highly with academic achievement and load on g. Put differently, they reflect self-regulated learning. Additionally, intercorrelations among inductive tasks differ with respect to content area and task demands.

In a review of inductive learning, Simon and Lea (1974) attempt to eliminate the distinction between the rule induction and problem solving literatures. The basic premise is that problem solving not only involves but <u>is</u> inductive reasoning. The authors define problem solving as a search through a problem space of knowledge states until a state provides a solution. Again the critical elements are in connecting nodes that contain information relevant to the problem with each subsequent node providing additional data to the knowledge state. Problem solving is characterized in terms of a problem space containing knowledge states that act as input and generate new knowledge states. Selective search leads to inferences and hypothesis testing analogous to the inductive processes of rule discovery. According to the authors, the only difference between problem solving and rule induction is the test process. Tests in problem solving are comparisons among pairs of knowledge states, whereas in rule induction, the tests are made by applying a rule to the

given objects and assessed to determine if the application is correct. The test is part of an indirect and iterative path toward a final solution.

Problem Solving and Knowledge States

The distinction between problem solving and inductive learning remains unclear. The two may be one in the same or separate but closely related. Both appear to involve task focus as we have defined it. The literature on problem solving does address several issues critical to the present review. Specifically, it provides a study of the research on knowledge states, strategic planning, and the training of problem solving methods that links to the differential and instructional psychology literatures as well as to CAI.

Second, the identification of cognitive processes relevant to and the training of problem solving provides an integrated perspective from which to examine effective instructional procedures, whether in the traditional classroom or computer environments. Several approaches to the research problem exist. These range from using inquiry teaching (Collins & Stevens, 1981; 1982) to providing explicit sequences of plans (Corno & Collins, 1981; Polya, 1957) that attempt to optimize performance. Research on training methods for problem solving has produced many programs, yet little evidence clarifies the cognitive impact of problem solving (Larkin, 1980). Resnick and Glaser (1976) provide a definition of problem solving that underlies the concept for the purposes of the present review:

> Psychologists agree that the term "problem" refers to a situation in which an individual is called upon to perform a task not previously encountered and for which externally provided instructions do not specify completely the mode of solution. The particular task, in other words, is new for the individual although processes or knowledge already available can be called upon for solution (p. 209).

Training Problem Solving Skills

Polya's book (1957) remains the seminal work on problem solving instruction -- or how to organize information and plan toward a solution. According to Polya, the first task for a student is to understand the problem. Understanding involves finding suitable notation for the problem and producing some sort of representation. Devising a plan is the second step and depends upon how students encode the given data and all the attributes. It is necessary to discover some link between the known and unknown information. The next phase consists of carrying out the plan and monitoring progress at each decision point. The final step is a comprehensive examination of the solution and potential applications to other tasks. Thus, successful problem solving for Polya rests on careful examination of details, the development of a general outline of the problem space, and continuous monitoring. Students are seen as acquiring such skills through imitation, practice, and use of previous

experience. Instruction that provides too much or too little help will hinder learning; therefore, students must assume most of the responsibility for their own work. Students must become active learners who organize and impose structure on the instructional material and mobilize their cognitive resources to obtain successful solutions to the problems. This type of learning extends beyond mere task focus to encompass all five self-regulation activities.

Corno, Collins, and Capper (1982) present a similar sequence of classroom instruction for improving motivation and academic performance in low ability students. One part of this system involves directly teaching self-regulation skills. Using an inductive modeling approach, students are taught to approach a variety of school tasks with a five-step "Learning Formula." Second, the first step is to focus on the task and determine what the task requires. The task is broken into component parts while students decide what facts are important to discover. Third, students then must relate the present task to previously solved, similar ones, and use that prior experience in the current situation. The fourth step consists of making a reasonable plan toward a solution. Finally, students check their work to monitor progress. Again, successful learning is based on careful analysis of the problem's components, relating these data to what the student already knows, making a plan, and continuously monitoring progress, i.e., the skills of self-regulation.

These two approaches to problem solving resemble what Larkin (1980) calls a means-end analysis. This involves determining the appropriate operations that minimize the difference between present knowledge states and knowledge required to obtain a solution. Larkin comments that such an analysis is a generally effective, yet inefficient strategy, which is used when more efficient methods are not available. Expert problem solvers, in particular, use creative variations on means-end analyses, sort of "doing their own thing." Thus, teaching sequenced planning to solve problems to high ability learners could interfere with their own effective processing (Salomon, 1979).

A slightly different approach to training problem solving is that of inquiry teaching (Collins & Stevens, 1981; 1982). The approach incorporates the goals and subgoals used by effective teachers, and the strategies and control structures needed to obtain those goals. The authors outline 10 strategies seen as integral parts of inquiry teaching. First, the selection of positive and negative exemplars are used to highlight relevant features of the task. Sequences of examples are varied systematically to elicit either differentiation or generalization among factors. A third strategy is the selection of counter-examples, and a fourth concerns the generation of hypothetical cases. Examination of hypothesis formation, evaluation, and the consideration of alternative predictions are three other strategies. Entrapment involves forcing students to acknowledge and correct misconceptions while tracing the consequences of the errors to specific contradictions in reasoning. The final strategy emphasizes the importance of questioning authority rather than passively accepting all instruction as fact.

Ability and Strategic Differences in Problem Solving Performance

Collins and Stevens focus on the types of teaching strategies that facilitate problem solving, while Sternberg and Davidson (1982; 1983) emphasize the relation of ability differences to performance. Sternberg and Davidson identify three general processes required for successful problem solving: (a) the ability to select and encode information relevant to the particular problem, (b) the ability to integrate unrelated information, and (c) the ability to compare the current problem to previously encountered ones. The authors examined the relation of induction and deduction to ability and found that performance was influenced by the type of reasoning required. Both induction and deduction were significantly related to ability, but the relation was much stronger for induction. High ability and successful problem solvers required more time to complete tasks than did lower ability students. The low ability learners selected obvious but incorrect answers impulsively or did not know how to begin work and therefore gave up. Using the language of our framework these learners would probably have tried resource management if the task had permitted it. Sternberg and Davidson (1982) conclude the following about task approach and cognitive types:

> People approach problems with different cognitive styles. In particular, some tend to be more impulsive and others more reflective. It seems to us -- although we have no hard experimental evidence to support our view -- that to combine both impulsive and reflective styles (p. 44).

Johnson (1978) examined strategic variations on concept learning tasks across learners of different abilities. He distinguished three types of strategies. First, tacticians approached the problems with high efficiency, yet a low rate of response. Scanners showed a high rate of response but were low in efficiency. Focusers were efficient but also responded at a fairly high rate. Johnson found that focusers and tacticians were the high ability students while the scanners were the low ability learners. Apparently both cognitive strategies and cognitive styles influence performance on problem solving tasks. However, as Messick (1982; 1983) notes, cognitive strategies are more amenable to change and training and therefore perhaps should be the focus of instructional efforts rather than cognitive styles. Yet, cognitive styles influence how students approach a given problem but knowledge and knowledge states are the quintessential factors, not only in problem solving but in learning in general.

Greeno (1980b) points out that problem solving is knowledge-based, with knowledge taking the form of general strategies for analyzing tasks and attempting solutions. Such strategies involve procedural knowledge that allows for organization, reduction of processing burdens, integration of relations among objects, setting goals, and planning toward a solution. Planning toward a solution is seen as the application of task constraints and procedural rules used to transform and progressively refine goals to subgoals. VanLehn and Brown (1980) view planning knowledge as "not only the functional decomposition of the surface structure of the procedure but also the reasoning used to transform the goals and constraints that

define the intent of the procedure into its actual surface structure" (p. 107).

Knowledge States

Rowher (1980) and Greeno (1980) specify the type of knowledge thought to be critical to effective problem solving. Greeno (1978; 1980a) uses the term strategic planning knowledge, while Rowher (1980) uses elaborative knowledge. According to Rowher, effective learners form relationships among problem elements by elaborating them into mental events. Elaboration is activated by indirect prompts that range from explicit to implicit. The effectiveness of such indirect prompts differs across learners and depends on prior knowledge. Evidence indicates that older and more able learners need only implicit prompts to stimulate elaboration, while younger and less able students require explicit prompts (see also Reder, 1980).

Another line of research approaches the relation between the explicitness of instruction and learning. Brown, Collins, and Harris (1978) examine explicitness of cues in artificial intelligence and find that simply alerting students to the existence of a few simple strategies can provide beneficial information. Awareness of strategic planning knowledge apparently is the critical element. Yet, presenting material as clearly and explicitly as possible can be counter productive for long-term goals -- particularly when recipient learning is permitted -- for students will not gain full understanding of the concepts but only master them for immediate tasks. In contrast, if students struggle to understand an idea, short-term performance may be affected negatively, but the long-term outcome should be improved comprehension. The authors present a compromise to the explicitness problem.

> If a concept is explained without explicit reference to the complex processes necessary to understand it, then the student will not be able to reconstruct the process alone. However, if a concept is presented by showing successively how to refine one's understanding of the concept (or, more metaphorically, how to experiment with the concept in order to "debug" one's own understanding of it), it will not be counterproductive (p. 109).

The distinction being made here is between knowledge needed for problem solving and knowledge needed for understanding. Knowledge needed for successful problem solving may benefit more from explicit than implicit instructional support although implicit cues may aid some learners. Knowledge needed for understanding depends heavily on implicit instruction for maximal long-term benefits. Put differently, an effective instructional strategy would systematically model and gradually require students to overtake the problem solving process.

Top-down and bottom-up processes. Greeno (1978), as well as Brown, et al., (1978), acknowledge the differential influences of top-down and bottom-up processing on problem solving activities. Top-down processing

consists of general hypotheses that direct cognitive activity, whereas bottom-up processing involves using the given information to make decisions and generate new hypotheses. Both processing types involve the acquisition and reduction processes that characterize self-regulated learning. Problems generally require both types of processing. Brown, et al., define such a combination top-down/ bottom-up processing. Planning knowledge and hypothesis formation and evaluation processes are considered the top-down component. Processing hypotheses, confirmation, and revision processes form the bottom-up component. The bottom-up component may be seen as critical to understanding; the top-down component may be more essential to problem solving.

Strategic planning knowledge. There is a specialized form of knowledge that is hierarchially organized and combines both top-down and bottom-up processing, which appears critical to problem solving and general understandinng. This is Greeno's (1978; 1980a) strategic planning knowledge. Strategic planning knowledge is seen as the knowledge that organizes activity on a task, sets goals, and forms plans that lead to goals. It involves the integration of existing information with incoming information, identification of structural relations among problem elements, and continuous monitoring or progress. Furthermore, strategic planning knowledge involves simple goal setting that leads to a subgoal structure of increasing complexity (Greeno, Magone & Chaiklin, 1979). Planning toward a goal includes knowlege of actions, preconditions and consequences of actions, and components of subactions that combine toward the more global action. Goal setting depends upon how well a given problem is structured (Greeno, 1976). A well-structured problem is one in which specific elements, rules, and operators can be applied to form a plan of action toward a goal. Greeno's analyses indicate that not only do definite goals result from poorly structured problems but also that such goals are critical in solving well structured problems. An indefinite goal here is seen as a pattern recognition system that identifies alternative routes toward an ultimate solution. The top-down/ bottom-up conceptualization of processing is the predominant mode here, since hypothesis formation and elaboration, as well as proposal, confirmation, and revision are essential components in the progression toward a final solution. These concepts and strategic knowledge are not taught explicitly in the classroom, although students acquire it by induction, possibly through sample problems. Greeno (1980a) states that:

> The principles of strategic planning knowledge that must be applied in solving problems probably can be induced as general properties of the sequence of steps that students observe in example solutions. Knowledge that is induced in this way is implicit in nature (p. 5).

Apparently, strategic planning knowledge is domain-specific, and can be taught explicitly or by discovery learning. Greeno also warns that explicit instruction in strategic planning knowledge may be hazardous because it stimulates passive learning (or fosters recipience), thereby decreasing the educational benefits gained from finding one's own solutions. The author instead emphasizes the need for active generation of domain-specific strategic plans.

Glaser (1982) echoes the need for a better understanding of strategic planning knowledge, and its relations to the explicitness of instruction. Glaser summarizes with the following statement:

> This strategic knowledge is usually relegated to the students' general ability to apply what is actually taught (i.e., intelligence). It is possible, however that such strategic problem solving, if it can be analyzed and understood, could also be explicitly taught. Thus, a new problem for instructional research is to study the nature of this skill and to investigate ways to teach the strategies involved so that they are fostered in students who are able to readily induce them. Research of this kind is now seen as an important aspect of mathematics instruction and of studies in problem solving in general (p. 297).

Planning

Glaser appeals for a comprehensive understanding of strategic planning knowledge as it is subsumed under more general academic and problem solving skills. However, few research endeavors have undertaken this task, and in particular, have focused on planning as a cognitive skill. Apparently, the main thrust of the literature emphasizes executive planning in relation to business and strategic planning in the sense of military applications. Perhaps Yinger (1977) provides the most comprehensive review of the planning literature. However, the seminal work on planning as a cognitive skill is found in Miller, Galanter, and Pribram (1960) who state that "a plan is any hierarchial process in the organism that can control the order in which a sequence of operations is to be performed (p. 16). In general, planning is a decision-making process, oriented toward attainment of a future goal. In a manner similar to Polya's (1957) problem solving work, planning consists of four component skills: (a) setting goals, (b) forming alternatives, (c) predicting outcomes for the alternatives, and (d) evaluating the alternatives in relation to the goals and outcomes (Yinger, 1977).

Furthermore, Yinger notes that most planning requires productive thinking in that the given problems are usually novel and unique. However, reproductive thought or past knowledge also influence planning processes. Similarly, Eastman (1970) distinguishes between "top-down" and "bottom-up" planning. "Top-down" planning is based on past experience whereas "bottom-up" planning aggregates subgoals or elements in a hierarchical design until the ultimate goal is reached. In either case, planning is closely connected to problem formulation. First, learners must recognize the problem and goals and translate them into an internal representation. They then must define the problem space and construct a sequence of subgoals in a search for the solution. Actions are taken, the results evaluated, and an iteration of subgoals undertaken to approximate the end state more closely. Iterations of problem formulation and immediate solving processes eventually converge on the ultimate goal state.

Planning in Artificial Intelligence

McDermott (1978) designed a computer problem solver that addresses this iterative notion. Problem solving entails identifying subproblems until they are at a solvable level. These subproblems results in a plan that must be executed to attain the solution. The network of subtasks is the plan. Tasks are reduced to a plan schema while choice rules aid the learner in determining the most appropriate plan for the given problem.

Similarly, Wilensky (1981; 1983) conceptualizes planning in terms of high level goals and plans refered to as meta-goals and meta-plans. Problem solving also is seen as a sequence of operators that transforms the learner from initial to goal state. However, most problem solvers lack an understanding of goals, inferences, and the general planning process. Such knowledge about planning is meta-planning and should be expressed in terms of a set of goals (meta-goals) and plans (meta-plans). Wilensky has designed a planner based on these premises. The planner consists of a goal detector, plan generator, projector, and executor. The author (1983) outlines several requirements of a planner:

1. Plans are associated in memory with the goals to which they apply.
2. Plans that are associated with a particular goal can be retrieved from memory by specifying that goal.
3. The planner can project plausible hypothetical features from its knowledge of the present world together with its own tentative plans.
4. Goals can be inferred based on the situation in which a planner finds himself.
5. The planner must be capable of detecting the interactions between its goals.
6. These interactions must be taken into account in its subsequent planning processes.
7. In addition to generating and modifying plans for goals, the planner must be capable of evaluating alternative scenarios and of abandoning some goals in order to secure others (p. 19-20).

The planner also must consider positive and negative relationships, goal conflict, and goal competition.

Cohen and Feigenbaum (1982) review general planning and problem solving in relation to artificial intelligence. Efficient problem solving relies on planning processes since plans monitor progress and identify errors en route to a solution. The authors also note that plans reduce search, help to resolve goal conflict, and provide for error recovery.

Furthermore, there appear to be four approaches to planning in artificial intelligence. First, hierarchical planners consist of a hierarchy of representations. At the highest level is an abstraction of the plan, while at the lowest is a detailed plan, sufficient for problem solving. Nonhierarchical planners contain a sequence of actions in which means-ends analyses reduce goals to simpler ones. One problem with this approach is that it fails to distinguish between critical aspects and unimportant details within the plan. In contrast, script-based planners do not generate plans but rather store and recall plans for future use. An appropriate plan for a given problem is recalled and the details fitted in with operators for the particular problem situation. Finally, there are opportunistic planners that are multidirectional in nature and contain a bottom-up component, driven by opportunities in the developing plan. Much of the artificial intelligence work had focused on the hierarchical approach whereas human planning and cognition have been emphasized in the opportunistic model.

Human Planning Processes: The Opportunistic Model

Hayes-Roth (1980) and Hayes-Roth and Hayes-Roth (1978; 1979) present a cognitive model of opportunistic planning which involves two primary processes. First, planning is seen as the formulation of activity toward attaining a goal. Second, control focuses on monitoring and guiding the execution of a plan to its conclusion. Opportunism differs from the iterative planning models in that current decision points suggest opportunities for further plan development. Opportunistic planning therefore is conceived as multidirectional in that the "top-down" component guides decisions from abstract to more detailed plans while the "bottom-up" component begins with detailed plans that stimulate higher level abstract plans. Furthermore, opportunistic planning is incremental because decisions at any point in time eventually are integrated into a final complete plan. Lastly, the authors view their model as heterarchical rather than hierarchical.

The opportunistic model rests on the notion that there are independent plan specialists that are part of a decision structure, called a blackboard. The blackboard, allowing for interactions among the specialists, is separated into five planes that reflect conceptual categories of planning (see Hayes-Roth & Hayes-Roth, 1978; 1979). Briefly, these planes include (a) metaplan decisions-general approach to the problem; (b) plan decisions-intended operators to the problem; (c) world knowledge decisions-enviromental information that might influence plan decisions; (d) plan abstraction decisions-desirable actions; and (e) executive decisions-organization of the planning process. Each plane contains several levels of abstraction at which the specialists generate tentative decisions and function.

Empirical tests of the model indicate a number of findings about planning processes in general, and more specifically how to improve planning performance. Subjects made decisions at all levels of abstraction and planned opportunistically rather than hierarchically. They also were able to transfer strategies used on other planning tasks and could adopt specific strategies in response to explicit instruction.

Good and poor planners were differentiated by three types of planning behavior. Good planners used all parts of the blackboard while poor planners less widely made use of the blackboard. Good planners used more planning specialists and exhibited more attentional flexibility than did poor planners. Additionally, such factors as goals under consideration, time available, and difficulty of achieving goals given the time constraints influenced planning performance. Finally, students consistently underestimated the time needed to complete the planned tests.

These findings resulted in several suggestions for improving planning performance. The authors recommend three training methods: a computer simulation of effective planning, induction of illustrative experiences, and explicit instruction. These methods should stress domain-specific planning heuristics, the use of multiple levels of abstractions and all decisions planes, general planning strategies, and cognitive strategies for resource estimation. Large-capacity memory, attentional flexibility and strategic flexibility are cognitive skills implicated as important to effective planning.

Computer Applications and CAI

Computers have become increasingly powerful problem solving tools in science, medicine, and other disciplines as well as educational tools that provide an alternative instructional medium to the traditional classroom. Microcomputers are a new and exciting domain for learning in which students become active and self-directed participants rather than passive observers in the educational process (Papert, 1980). Student-computer interfaces vary in the degree to which students assume responsibility for their own learning. There are instructional materials that completely control the learning situation; the computer "programs" the student. For example, programmed instruction with or without a computer objectifies the learning process, and divides it into manageable steps, but does not permit students an active role. Although instruction is based on individual response patterns and provides instant feedback, students are still passively dependent on the computer program. Learning is recipient. Such programmed instruction without computers was the first of four phases in the evolution of CAI (Goldstein & Carr, 1977).

The introduction of computers marked the start of the second phase, where instruction included decision trees of items with branching dependent upon student responses. According to Goldsteain and Carr, the goal of this phase was "to develop tutor languages to facilitate the design of scripts for teachers for their domains" (p. 228). The third stage of development introduced the use of domain expertise in CAI tutors. WUMPUS (a complex problem solving game in CAI format) is one such program that had its beginnings as a simple expert tutor (Carr, 1977; Stansfield, Carr, & Goldstein, 1977). The current developmental stage extends and applies CAI tutors to a theory of learning and teaching. The expert tutors now include models for learning and tutorial strategies. Such modeling of students' knowledge is called AICAI since artificial intelligence is applied to the computer-assisted instruction.

Computer Tutors and Coaches

Barr and Feigenbaum (1982) outline three approaches to computer applications in education that are apparent in research on intelligent tutors. First, students can learn problem solving methods through direct interaction with the computer. Second, games or simulations can be used as instructional tools. And finally, formal CAI can be administered. There are several types of tutors or coaches based on these perspectives. Computer coaches provide advice within an athletic paradigm of skill acquisition (Goldstein 1976; Goldstein & Carr, 1976), and encourage problem solving abilities by actively involving a student in the learning environment (Barr & Feigenbaum, 1982). The first coaches were script-based CAI in which diagnosis of correctness was limited to the completeness of a script containing appropriate responses. Limitations occurred because of too frequent or inappropriate interruptions that decreased the learning and motivation potential of the programs. It was necessary to make the tutor less obtrusive and more "perceptive." Expert-based CAI followed with hypotheses concerning the expert knowledge and skills necessary for successful performance on a given task. But these systems assumed incorrect answers reflected a lack of skill inherent to the beginning learner, rather than variations or simplifications of experts' performances (Goldstein, 1979).

This difficulty led to a learning overlay model in which the tutor determines the presence or absence of specific skills. Students' understanding is represented in terms of an expert component that reflects how a skilled individual would perform in specified situations. Students' responses are seen as deviations from or variations on the expert's performance. A matching pattern recognition procedure between student and expert responses is used to assess response histories and flag errors within the network (Barr & Feigenbaum, 1982). Goldstein and Carr (1977) developed a four part overlay model for the WUMPUS program using a production-rule representation of knowledge states: (a) the Expert uses rules of skill, (b) the Psychologist uses rules of evidence, (c) the Tutor uses rules of explanation, and (d) the Speaker uses rules of language. The overlay model then was refined into what Goldstein (1979) calls a genetic graph of skills. Procedural rules are represented by nodes and links, with specific planning knowledge rules grouped into islands, oriented to a particular goal. The coach tutors in terms of these islands and gains power from the hierarchical structure of the grouped rules. Goldstein points out that the genetic graph directs a tutor by: (a) refining a structural model of the knowledge state, (b) providing a metric for skills, and (c) structuring a model of learning behavior.

A final type of coach takes the "bug approach," which assumes that novices and experts differ and therefore the former's performance is modeled as a deviation from that of the expert. Brown and Burton (1978) describe a model of bugs or mistakes in basic mathematical skills. They identify not only what mistakes students make but also why such errors are likely to occur. The underlying premise is that mistakes follow incorrect procedures; therefore, both the correct and incorrect procedures must be modeled. Buggy subprocedures replicate errors, underlying causes, and provide specific remediation. A computerized game called

BUGGY has been developed to diagnose the interactions between students and tutor using a procedural model of skills and subskills.

Educational Games and Simulations

Burton and Brown (1979) defend the use of games and coaches in learning environments. They state that games are good problem solving environments that stimulate explanation, discovery learning, and formulation of strategies and knowledge structures. The role of the coach is to guide discovery learning, prevent gross errors, and allow the student the fredom of decision and observation of results through perceptive yet unobtrusive feedback. Not all games are appropriate, nor do they provide maximal learning experiences for all students (Malone, 1980; 1981; Malone & Lepper, 1983) Certain characteristics of a computerized game must be apparent for it to be intrinsically motivating and appropriately educational. Games should contain challenge, intrinsic fantasy, and stimulate curiosity and control. An optimal level of challenge that depends on goals with uncertain outcomes is a necessary inclusion. Decrements in motivation result if the game is too difficult whereas it becomes unintereting if too easy. Challenge is maintained by including variable difficulty levels, multiple-level goals, hidden information, and randomness. Goals must be personally meaningful, obvious or easily generated, and should include performance feedback. Fantasy varies with individual preferences, but if intrinsically generated, it may increase students' interest in the gaming situation. The game also should stimulate students' curiosity by highlighting inconsistency, incompleteness, and a lack of parsimony in examples and by providing just enough information to make existing knowledge seem incomplete, thereby motivating students to respond to feedback and complete the cognitive structure. Finally, the game should allow students to assume some control and power over the learning environment. Providing some choice of self-determination gives the students a perceived sense of control over the learning process.

Malone and Lepper's (1983) taxonomy of intrinsic motivation also considers interpersonal motivational factors such as the need for cooperation, competition, and recognition. Although research exploring the taxonomy was conducted on educational computer games (Malone, 1980; 1981), Malone and Lepper (1983) and Lepper and Malone (1983) stress that the instructional principles can be extended to other types of computer-assisted instruction as well as more traditional instructional media.

Orbach (1977; 1979) and Bredemeier and Greenblat (1981) consider similar taxonomic characteristics of simulation games that make them educationally effective. A strength of simulations is that their resemblance of real life contributes to realistic practice of skills that may otherwise be impossible. This tie to the real world necessitates a careful balance between inclusion of too many or too few details (Orbach, 1977). If too few details are included, the simulation will fail to recreate the real world and therefore cease to be a simulation. Conversely, if there are too many details, distractors will encroach on learning and the simulation will cease to be a model.

Furthermore, simulations engage students actively by providing the unique opportunity to role play in an environment that normally may be inappropriate for practice and knowledge acquisition and application. Other important characteristics of simulations are that they provide clear objectives and criteria for winning, immediate and riskless feedback, and a line for multidirectional communication (Orbach, 1979). These characteristics, in addition to novelty, surprise, competition, and curiosity all contribute to the motivating nature of simulation games.

Issues Affecting Implementation of CAI

The role of the learner. Lepper and Malone (1983) note that the learner's role in computerassisted instruction, whether games or simulations, varies according to the part of the taxonomy stressed by the particular program. Learners become problem solvers with emphasis on challenge, information processors with curiosity, and volitional actors when control is the focus. The issue of control is one which is very much a source of debate between the artificial intelligence researchers and the proponents of framed-based CAI (McCann, 1981; Miller, 1983). Merrill (1980) argues for the inclusion of learner control in CAI while others, including differential psychologists such as Snow (1980b) oppose giving students complete control over their learning environment. Snow's argument is that students are incapable of making informed decisions about the content and sequence of instructional materials and therefore would suffer educationally if given control over what, how much, and when to attempt certain tasks. Part of the control issue is what Lepper and Malone (1983) refer to as the behavioristic "minimax" strategy. If students are given too much control, they are likely to take an instructional course of least resistence, thereby missing exposure to the full range of instructional materials. In contrast, Merrill purports that students are capable of making such informed decisions and therefore should be provided with some choice over their learning activities. DiSessa (1981) also stresses the students' need for an education in which they have a sense of control, are allowed to set goals, and develop long-term links with the materials. In general, there is disagreement in the field as to the degree to which students should assume control of their own learning and where that locus of control should reside (Malone & Levin, 1981).

Design considerations. A number of other issues at the heart of CAI design have important implications for instruction. CAI can create a response-sensitive learning environment. One question is how best to integrate hints without undermining motivation and learning (Burton & Brown, 1979; Brown, 1981; Lepper, 1982; Lepper & Malone, 1983). Similarly, Atkinson (1972) stresses the need for response-sensitive instructional material that takes into consideration a student's response history in making subsequent instructional decisions. Feedback, in the sense of being response-sensitive also should be ipsative in nature for low ability students and normative for high ability learners. Design principles such as the inclusion of randomness, multiple level goals, and variable difficulty levels can interfere with learning and transfer of skills if not integrated appropriately into the instructional materials (Lepper & Malone, 1983). Misuse of these factors could result in

incongruence between the goals of the program and those of more general curriculum, attentioinal focus on irrelevant rather than relevant content, and inefficient use of learning time. Lepper and Malone also question the effects of making educational activities fun on other instructional media, learning in general, and the transfer of skills.

Differential access. One issue that is related both to immediate instructional practices, but more critically to long-term educational practice is one of differential access to CAI. Access to microcomputers varies across the country and even within local areas. The fear is that the rich will get richer and the poor will fall further behind. Particular groups also may suffer from differential access due to such reasons as interest and parental support. Kiesler, Sproull, and Eccles (1983) discuss some of the implications for education and career options based on patterns of computer access for girls and boys. Hess and Miura (1983) focus on gender and socio-economic differences in enrollment in computer camps and classes. Results indicate limited enrollment of girls, minorities, and poorer students. These patterns become more apparent with increasing age of students and more advanced curricula. These issues raise important questions about the implications of computer use for different groups of students.

AI vs. education: The issue and related research. Another issue that underlies the use of computers in schools is one of a theoretical nature. CAI has been criticized for lacking any theory of learning (Hall, 1978) or failing to specify a pedagogical basis for instructional activities (Lepper & Malone, 1983). If these criticisms are well founded, perhaps they may be due to the backgrounds of those who control the computer industry. For the most part, input from educational, instructional, and psychological experts has been lacking, leaving important pedagogical decisions to computer scientists, engineers, and programmers. Some of the differences in the perspectives of psychologists and AI researchers have been delineated by Schank (1980). Addressing an audience of differential psychologists, Schank notes that there has been no interaction between researchers studying intelligence and artificial intelligence. He even goes as far as to say that AI researchers are not interested in intelligence. Rather, they are more concerned with the "ways intelligence manifests itself in a cognitive system" (p. 5). Furthermore, the focus of research should not be on problem solving or reasoning. Memory, the memory processor, and knowledge store are critical to the understanding of memory organization, input, and control of inferencing in AI. According to Schank, the determinant of intelligence is the ability to distinguish relevant from irrelevant input in terms of future processing needs.

One attempt to integrate the computer science literature with cognitive and educational psychology is evident in a review by Mayer (1981). In examining how novices learn to program, Mayer considers numerous studies that relate educational and psychological theories to computer programming instruction. Apparently studies that examine cognitive and affective factors, differential and instructional variables, and problem solving skills by using software and computers as both research tools and a source of experimental interest are more prevalent means of integrating the disparate literature and research perspectives. Such studies

not only use the computer as an investigative method but more importantly caapitalize on the unique characteristics of computer environments for learning and examine the cognitive demands and consequences of computer-assisted learning.

Kahn (1981) examined the effects of instructional variations and individual differences in cognitive and affective variables, on performance on a computerized problem solving game, called Game X. The main problem solving skill of interest was strategic thinking. Results indicated that the role of individual differences in affective and cognitive variables differed across instructional groups and task difficulty. As task difficulty increased, cognitive variables became more important predictors of performance whereas affective factors were better predictors in easier versions of the game. Furthermore, pretraining in relevant problem solving and strategic thinking skills was more effective for some but not all students.

Cox (1980) examined the interaction of learner characteristics on problem solving strategy and students' performance on three computerized deductive reasoning problems. Training on a 10-step, problem solving model also was tested in this study. Results indicated the experience with the computerized problem solving tasks improved problem solving skills, but that training on a specific strategy did not improve performance (as measured by latency). However, students who received strategy training were more likely to use the strategy on similar tasks. Therefore the training was at least successful in introducing a new and useable strategy. Furthermore, students working in groups were more successful than those working alone. Group interaction allowed students of all ability levels to solve the problems, and maintain interest and motivation across experimental sessions.

Seigner (1980) studied the differences between game and academic ability using a non-zero sum simulation game, called Generation Gap. Apparently, game and academic ability differ, as reflected by correlational patterns. Cognitive variables affected both abilities in a similar manner and were stronger in general than the affective factors. Affective variables had less influence on game than academic ability. Game ability, as required by the specific abstract problem solving task, is similar to academic ability in that it is influenced more by cognitive than affective variables.

In one final example, Mandinach (in progress) has examined a computerized game called WUMPUS in an attempt to integrate the literatures of differential and instructional psychology, problem solving, planning, and artificial intelligence. WUMPUS attempts to activate student self-regulation when other instructional support is not provided. The present instructional version of WUMPUS is neither a coach/tutor nor typical of traditional CAI. It does, however, combine Barr and Feigenbaum's (1982) three applications to education and Malone's characteristics, particularly challenge, to create a stimulating learning environment. The intent here is to incorporate an intrinsically motivating game as an instructional tool into a CAI paradigm whereby students acquire problem solving methods or strategic planning knowledge to become self-regulated through direct interaction with the computer. WUMPUS support materials differ from

existing CAI in that they do not provide direct instruction or tutoring in the traditional senses. Rather, the WUMPUS programs provide cued examples from which students are expected explicitly or implicitly to induce strategic planning knowledge. WUMPUS is not meant to be the definitive CAI tutor nor an "optimal" program for instruction in self-regulation or strategic planning skills. The WUMPUS support materials are, however, an attempt toward integrating four disparate literatures and applying their findings in one comprehensive research endeavor. The exploration of the effects of cued instruction through examples on learners of different abilities provides the needed link between problem solving, instructional methodology, and the man-machine interface.

The present review has attempted to provide a comprehensive survey of the literature seen as relevant to computer assisted learning. It is apparent that little research focuses specifically on the cognitive demands and consequences of computer environments for learning. However, each of the five separate literatures discussed provide results that are critical to the study of computer assisted learning. These literatures, taken in combination, provide a convenient perspective of the instructional, learner characteristics, cognitive, affective, and software design variables that are important to present and future research efforts in the area. Perhaps an integrated approach to the cognitive factors involved in computer enviornments for learning will one day become a reality. Until that time, it is our hope that researchers in each of the disciplines will recognize the worth of the other approaches and attempt to integrate some of the findings into their own work.

CHAPTER THREE

Research Applications: Characteristics of Software

Introduction

The validity of the ACCCEL Project's Series B studies relies heavily on the extent to which the software selected reflect the existing types of programs available for use in both the home and school. Additionally, it is important that the particular pieces of software selected for examination stimulate the use and acquisition of higher cognitive skills. The amount and variety of software products continually are increasing at a staggering rate, thereby making a broad based review of educational software an impossible task. Rather, it is the purpose of this chapter first to explore briefly the dimensions on which software vary and how those dimensions may influence the cognitive demands and consequences of learning from such software. Second, we will describe the particular pieces of software selected for study and provide rationales for how these programs may extend existing knowledge of the effects of learning from these classes of software.

Educational software, like all forms of instruction, is only as good as the medium, content, and intent of the material provided. Students will take from the instruction what they intend: no more and no less. The burden is placed on the program designer and classroom teacher to make informed decisions about the selection of software appropriate for given instructional purposes. Fortunately, the use of computers in educational practice is a somewhat novel but powerfully motivating tool and therefore can compensate to some degree for the many less than adequate programs currently in use. Yet, the dimensions along which software differ determine largely the extent to which the media make effective instructional material. These dimensions take into consideration the instructional purpose, techniques, structure, technical characteristics, and target audience.

Proposed Courseware Dimensions

Applicability, Structures, and Instructional Intent

Existing software evaluation forms and marketing brochures provide clues as to the characteristics educators and computer companies think are critical to educational products. Some of these characteristics are specific to computer learning environments while others are

generalizable across instructional media. A first general area of concern is the applicability of a program, given the targeted audience; the age, grade, or ability level and the program's corresponding degree of difficulty in adapting to these student characteristics. Perhaps the costs and copy restrictions of software and the number of students that a single disk or cassette can reach make variable difficulty levels across grades a desirable characteristic for a program. Similarly, the group instructional size for which a program is designed not only influences classroom management but also numerous cognitive variables as well. Again, perhaps the most cost-effective type of program is one which can be used both with an individual student or for all-class activities.

The structure of the program also influences the appropriateness of use in certain settings. For example, the degree to which a program is self-contained and thereby allows a student to operate it without assistance affects the role of the teacher and the cognitive demands placed on the student for independent learning. Additionally, if a program is self-contained and does not require adult supervision or excessive amounts of supplementary instructional material, then that program also may be appropriate for educational use in the home. Other factors that influence structure concern the physical operation and the instructional format of the program. One must ask how the software physically operates with respect to the necessary hardware, data retrieval, and system errors. Similarly, one needs to know how directions and feedback are provided and the method of presentation for the stimuli. Does the program include graphics, sound, or purely verbal material? Such structure variables are apt to have important effects on both the cognitive demands and consequences of instruction. As Cronbach and Snow (1977) point out, learners with different ability profiles learn better under different instructional methods.

Finally, and perhaps most important are the instructional purposes and techniques used in a particular program. These factors structure the learning environment and most certainly influence the cognitive demands that are required by a task and the cognitive outcomes, as well. For example, a program designed for the sole purpose of remediation will place different demands on a student than will one intended for enrichment. Consequently, the resulting cognitive outcomes are expected to differ based on the purposes of the instruction. In a similar manner, instructional techniques should interact with learner characteristics and therefore affect the degree to which certain individuals benefit from the type of instruction. Some of the commonly identified instructional techniques include games, simulations, problem solving, drill and practice, and tutorials. Task analyses of the cognitive demands of programs in the same content area but using different instructional techniques should vary markedly. Consequently, prerequisite skills are expected to interact with the cognitive demands of the task, thereby influencing the consequences of interacting within the various computer environments for learning.

In many respects, the above listed variables are not easily separated into independent dimensions. Rather, the interactions among the factors contribute to the uniqueness of the computer learning environments. Perhaps it is the uniqueness of each piece of software,

as the program and student characteristics interact, that make the cognitive demands and consequences of CAI such a compelling instructional medium and area of research.

General Dimensions of Software

However, it may be possible to form three general categories that characterize the cognitive demands for particular programs. A first category might be considered the degree to which students must assume responsibility for and control over their learning environment. At one end of the continuum, the program, teacher, and supplementary materials control the structure of learning as in drill and practice or programmed instruction. More interesting is when a student can interact with the program and in doing so, helps to structure the environment. Many simulations and games provide such freedom.

A second category involves the cognitive load placed on students. Research has found that ability profiles interact with tasks' cognitive processing demands (Snow, 1976a). Based on the literature, processing demands are bound to be lower in programs requiring rote memorization such as drill and practice and tutorials. The programs here perform much of the work while the student simply must supply a series of responses. Conversely, in many educational games, simulations, and problem solving programs, students must manage a heavy processing load, thereby activating numerous higher cognitive skills.

A final category concerns numerous technical aspects (the six distinctive features) of computer environments for learning as described in the original technical proposal (Linn, et al., 1982). While precision, consistency, and complexity will remain stable across varying kinds of software, it is expected that challenge, interactiveness, and the provision for multiple solutions will change. Complexity here refers to the sophistication of the learning environment rather than the difficulty level of the particular piece of software. Multiple solutions are possible in games, simulations, and problem solving tasks where flexibility in obtaining answers is many times a desirable quality. Conversely, in drill and practice programs, although there may be a number of possible response paths, there usually is only one correct solution.

Interactiveness refers to the continuous cycle between student input and computer feedback. This feature insures that students will be active rather than passive participants in the learning environment because computer response depends on student input. The cycle varies across software dimensions in the amount and explicitness of feedback provided. Feedback in drill and practice may consist of no more than a simple "correct" or "incorrect" response from the computer whereas a tutorial may provide a very explicit analysis and explanation of a student's answer. While explicitness is an instructional variable, feedback in a second sense is linked more to computer-oriented variables. For example, in a program such as a space simulation, every key press on the student's part reflects

cognitive activity. The computer then responds to each of those inputs. The cycle tests the limits of human reaction time with the responses at a very molecular level. The rapid-fire nature of the interaction contributes to the uniqueness of the particular type of program. Conversely, in some types of software, computer feedback is withheld until a student's entire sequence of responses is complete. Interactiveness here is more at the molar level.

Challenge is the sixth feature that varies across software dimensions. Malone (1981) found that challenge stimulates intrinsic motivation in computer games and depends on goals with uncertain outcomes, variable difficulty levels, hidden information, and randomness. Decrements in motivation result if the game is too difficult, whereas it becomes uninteresting if too easy. These characteristics also can be extended to other classes of software.

Rationale for Selected Software

Overview

Keeping these characteristics in mind, we have selected five pieces of software that consist of simulations, instructional games, and problem solving exercises. The selected software represent a general class of cognitive tasks that are enjoyable learning activities that also stimulate the use and acquisition of higher cognitive skills. These tasks focus primarily on problem solving and logical reasoning skills and have been selected in light of the emphasis on such higher cognitive skills.

For the most part, the simulations in which we are interested depend only slightly on knowledge of a particular real world situation. The programs require problem solving in a hypothetical world defined by the computer and are typified by their lack of firm grounding in a specific content area. Rather, these programs emphasize general problem solving and logical reasoning skills that would seem to be characteristic of a broad range of problems. We do not intend to imply that context is irrelevant to problem solving since Linn (in press) and Linn and Siegel (in press) have argued explicity that context is a serious consideration in the study of problem solving. Rather, we are suggesting that in the study of computer learning, a common feature of many of the most powerful and new pieces of software is that they focus on general problem solving skills. Therefore, it would seem appropriate for an evaluation on computer learning to focus on these same skills.

Three of the selected pieces of software may be seen as context-free in that they are not tied to any specific content area. However, we have included one simulation that is context-dependent and focuses on problem solving and logical reasoning skills within the domain of high school physics and a final program, within the context of abstract art that teaches scientific reasoning skills. The selected programs vary across the identified courseware dimensions and approach

the acquisition of higher cognitive skills from different perspectives. Three of the programs, WUMPUS, Patter Discovery, and Rocky's Boots, can be considered self-contained educational games or simulations with which supplementary instructional materials can be used. Another selection, Creative Play, is not just a single program but rather an entire curriculum of 25 discovery oriented problem solving activities. Finally, Yellow Light, a physics simulation is intended to accompany instruction on physical concepts such as acceleration and deceleration. Each of these programs will be examined in turn. First, a typical courseware evaluation will be provided as a summary of the software. A more complete description and analysis of the software then will be discussed.

Study B-1: WUMPUS

Program Title:	Instructional WUMPUS
Requirements:	Apple II with 64K, Two disk drives
Description:	Instructional "hunt the monster" game based on mathematical problem solving skills, simple probability, and rule based knowledge
Color:	N/A
Sound:	Silent
Graphics:	Low Res for instructional examples
Grade Level:	Junior High to adult
Accompanying Materials:	Program description and operating instructions
Managment Features:	One practice and four instructional sessions each consisting of 12 parallel games. Cued examples accompany each of the instructional games.
Other Features:	The WUMPUS master disk generates the program while the data disk is used for accumulating a photo log of all respones, including reaction time.
Subject Area:	Problem solving/logical reasoning.
Specific Topic:	Strategic planning knolwedge, self-regulated learning, and logical reasoning.
Prerequisite Skills and Activities:	Simple probability and mathematical problem solving skills.
Instructional Purpose of Program:	Enrichment, remediation, or standard instruction.
Instructional Techniques:	Game, simulation, and problem solving.
Appropriate Group Instructional Size:	Individual (primarily) but can be used in dyads.

WUMPUS

WUMPUS is a computerized "hunt the monster" game that is based on mathematical problem solving skills, simple probability, and rule-based knowledge (Yob, 1975). The student is a hunter whose goal is to track down and kill a mythical creature, called a WUMPUS, avoiding several hazards that impede safe movement through a warren of 20 interconnected caves. The skills expected to be critical to performance on WUMPUS are outlined in a task analysis found in Figure 1. A more detailed description of WUMPUS, its characteristics, and requisite skills now is presented.

According to Yob (1975), the warren takes the form of a squashed dodecahedron in which each cave is connected to three others via tunnels. Legitimate moves are those using connected pathways between caves while illegal moves are those that violate the connectness of the configuration. Movement through the warren is limited by the placement of immovable hazards that endanger the hunter's safety. It is important to note that much of the information concerning hazards is not explicitly stated in the directions. Rather, as one gains experience with the game, the rules become increasingly apparent.

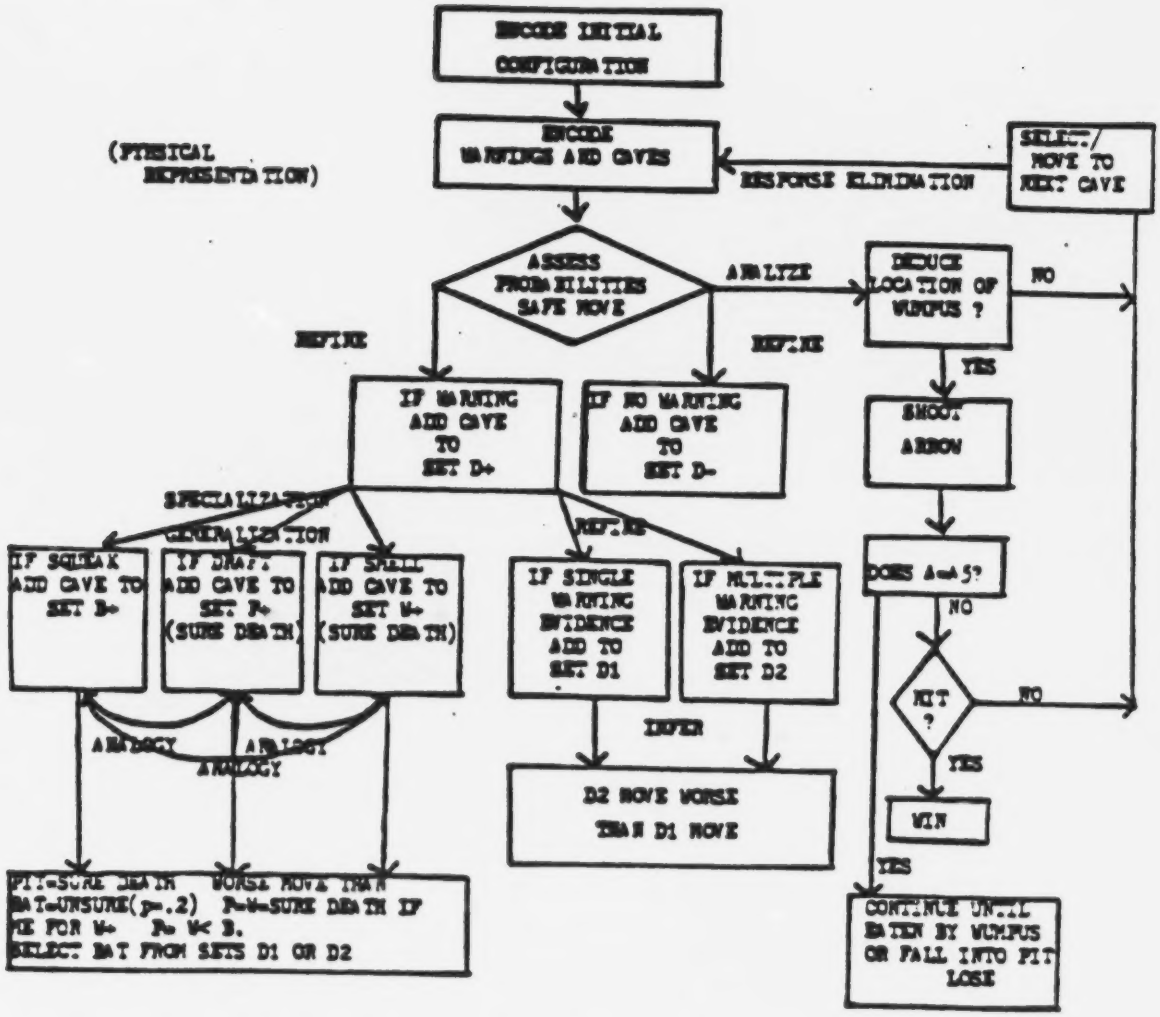
Goldstein (1979; 1980) and Goldstein and Carr (1977) conducted extensive research on WUMPUS, implemented a complex computer coach, and hypothesized about certain skills necessary to master the game. Goldstein (1979) presents a rationale for using WUMPUS in the study of learning:

> Becoming skilled poses a non-trivial learning experience for most children and adults: locating multiple dangers in a randomly connected warren of 20 or more caves can be complex. Hence, the game provides a useful problem domain for developing a theory of the evolution of procedural skills (p. 54).

Furthermore, complete mastery is mediated by the interaction of the student's ability to apply strategic planning knowledge within the task's rule-based constraints and the probability factors of the randomly generated configuration and placement of hazards (randomness will be controlled experimentally in order to increase a student's chance of winning). Regardless of how experienced one becomes with WUMPUS, there is always a chance of reaching a no-win configuration due to the location of the hazards. When that happens, no amount of strategic planning knowledge can save the hunter. These constraints create challenge and help to maintain interest in and motivation for the game. Conversely, because of the chance factors, students who lack ability always have some hope of winning. The uncertain outcomes are critical to maintaining the challenging and motivating properties of computerized games as well as in more traditional educational tasks (Malone, 1981). The present instructional version of WUMPUS minimizes some probability factors (e.g., making the WUMPUS immovable and controlling against presentation of hazard warnings in the opening configuration) in an effort to optimize the use of procedural skills.

Figure 1

Flow Chart for WUMPUS Task Analysis



The skills used in any particular game are dependent upon the randomly generated configuration. Some skills are dominant and are used continuously throughout the game while others are specific to particular classes of situations. First, the student encodes the initial configuration into which he has been dropped at random. Drawing a representation of the connections, warnings, and hazards within the warren is an important instructional aid. Probabilities of potential moves are assessed from the representation with the skilled student selecting safe over dangerous moves and refining and identifying the caves that belong to the danger sets. The danger set then is specialized into the bat warning, pit warning, and WUMPUS warnings. According to Goldstein (1979; 1980), analogies are formed between the types of warnings. (It is possible to construct a hierarchy of the degree to which certain warnings are more or less dangerous, relative to one another.) Further refinement of the dangers is possible based on the number and type of warnings presented. Multiple warnings provide evidence that a cave is potentially more dangerous than one with only a single warning. As the student amasses warning evidence, he is able to draw inferences as to the location of the hazards. Explicitly presented warnings assist the student to induce the location of alluded to but unstated hazards. The student selects a move that will help him to isolate the location of the WUMPUS, while keeping in mind rules, probabilities, and warnings. Careful analysis of the configuration and self-monitoring provide data as to problem solving progress. If a student has insufficient evidence to locate the WUMPUS or sometimes if his direct path is blocked by hazards, the iterative process begins anew. The student then selects another move and through a process of response elimination or mapping, adds new information to the configural representation, thereby progressing towards a solution. If the student believes he has located the WUMPUS, he then shoots an arrow into the identified cave. The student wins when an arrow kills the WUMPUS. The student continues to hunt the WUMPUS until he either wins, exhausts the allocation of arrows, falls into a pit, or gets killed by the WUMPUS. With experience, the student learns to take advantage of uncertainty and partial knowledge. Since an arrow can be shot through a sequence of five caves at a time, the student can use the uncertainty to make certain tests with each arrow. However, arrows bounce at random throughout the warren if the hunter defines an illegal path, creating the possibility that he may even shoot himself if the shot is attempted too prematurely.

Since the presentation of information unfolds with progress further into the configuration, it is critical that the student continuously monitors the accuracy of the data and representation. Such monitoring forces the student to recycle through the problem and identify implicit information from explicitly stated data and eliminated responses. Sometimes the configurations become rather complex, causing not readily apparent connections among the rooms. Errors from these non-obvious connections between caves are likely so checking increases the accuracy of the representation. Successful solutions require inductive reasoning in that the student must go beyond the given facts to draw inferences about the implicit and unstated information. Most games will lack complete configurations since the student either will be killed before obtaining a final mapping or will

succeed because the incomplete information will provide sufficient data from which to derive a solution.

Instructional WUMPUS

Two instructional versions of WUMPUS that differ in the explicitness of instructional support provided were designed. Both the high (guided modeling) and low (activating) intensity instructional treatments focus on particular and common errors or bugs that students are likely to make while processing a configuration. These bugs reflect basic reasoning deficiencies that refer back to the WUMPUS task analysis. Errors can occur anywhere within the task and relate to specific processes or strategies, but all reflect faulty strategic planning knowledge, self-regulated learning skills, and logical reasoning. The instructional version of WUMPUS is described in detail elsewhere and should be consulted for a complete discussion (Mandinach, 1983).

Study B-2: Rocky's Boots

Program Title:	Rocky's Boots
Requirements:	Applesoft, 48K, DOS 3.3, One disk drive
Description:	Simulation game using simple logical devices to build and test machines.
Color:	Yes
Sound:	Effective (On/Off)
Graphics:	Effective
Accompanying Materials:	Booklet with program description and operating tips.
Management Features:	Six levels that include a set of rooms with explanations, demonstrations, and exercises.
Other Features:	Supplementary material can be used to accompany instruction or discovery learning with the program. Can be controlled by the keyboard or a joystick.
Subject Area:	Logical reasoning
Specific Topic:	Symbolic logic, primitive logical elements, and rules of inclusion and exclusion.
Prerequisite Skills and Activities:	Basic understanding of such logical principles as AND, NOT, and OR.
Instrucitonal Purpose of Program:	Enrichment and standard instruction.
Instructional Techniques:	Instrucitonal game and simulation.
Appropriate Group Instructional Size:	Individual or small group.

Rocky's Boots is an educational game developed by the Learning Company of Portola Valley, California. The game presents an introduction to certain logical concepts and offers students a chance to incorporate these concepts into structures that are modeled as machines. The program is designed with six levels, each of which includes a series of rooms where logical elements are presented. By moving the cursor from room to room, students are introduced to the components of the game and given an opportunity for free experimentation. The fundamental components are wires, logical connecting gates, sensors, and a boot. Students must build a machine, given the available sensors, that will detect and kick specific objects. Several practice rooms are provided in which students can experiment with all sorts of machines.

The game becomes progressively more challenging as the number of elements increases. Active exploration of possible combinations of logical devices is encouraged. Students can work at their own pace by building machines, testing, and rebuilding as they choose. In the early stages of the game, the environment is carefully structured for the learner, whereas at more advanced levels, students can use the components to design and solve their own problems in a discovery oriented manner. Rocky's Boots therefore provides an open and highly interactive learning environment that is appropriate for students at various age, ability, and developmental levels.

Burbules and Reese (1983) have identified nine subtasks that are incorporated in Rocky's Boots. These subtasks are part of an initial approximation at a comprehensive task analysis. Research currently in progress should provide more detailed task analysis. First, students must interpret the task description. This includes identifying and comprehending such basic concepts as shape and color. Initial interpretation also includes an introduction to the rules of inclusion and exclusion. A second task involves analysis of the components in terms of a specified sequence of events. Students must attend to the stimuli in a particular order, thereby determining relevant from irrelevant concepts. Students must translate the sequence into available logic terms and then into machine language. Such translation requires the selection and planning of appropriate components.

A fifth subtask concerns visualization of the machine to be built. A visual representation should be a useful aid in the actual construction of the logical machine. Building the machine is the logical extension of the previous cognitive activity. The construction process is the physical output of the mental activity. Running the machine is a matter of testing whether or not the construction has been successful. The execution phase allows the student to monitor the machine's functioning. Debugging is necessary if the machine does not work properly. Debugging requires the student to realize that there is a malfunction and then isolate the cause of the error. Once the mistake has been identified, the student then must generate an alternative solution. The testing process begins anew.

Again, these processes are only approximate hypotheses of the higher cognitive skills required for successful performance on Rocky's Boots; The research currently being conducted should contribute to a more comprehensive task analysis and thereby a more complete understanding of the cognitive demands and consequences of learning from Rocky's Boots.

Study B-3: Creative Play

Program Title:	Creative Play
Requirements:	Applesoft, 48K, DOS 3.3, One disk drive
Description:	A curriculum of problem solving activities for the microcomputer that include games, simulations, and dialogues.
Color:	Yes
Sound:	N/A
Graphics:	Effective
Grade Level:	Ages 8 to 11
Accompanying Materials:	Comprehensive binder of program descriptions, operating instructions, and supplementary materials.
Management Features:	25 problem solving activities organized into 8 one-and-a-half hour sessions.
Other Features:	Activities are described in sufficient detail so that they can be changed to meet the needs of individual students or instructional purposes.
Subject Area:	General problem solving skills
Specific Topic:	Logical reasoning, data collection and evaluation, creative thinking, spatial visualization, pattern recognition, social interaction, and knowledge about computers.
Prerequisite Skills and Activities:	
Instructional Purpose of Program:	Standard instruction and enrichment.
Instructional Techniques:	Games, simulations, and problem solving.
Appropriate Group Instructional Size:	Individual or small group.

The purpose of the Creative Play curriculum is to introduce students (ages 8-11) to general problem solving skills, presented through a sequence of microcomputer activities. According to Friedamn and Slesnick (1982) who developed the curriculum for the Lawrence Hall of Science "it is a vehicle for recreation, a medium for generating visual and numerical patterns, a tool for simulating situations or experiments, and instrument for developing language skills, and an ordered collection of electronic parts" (p. 2). An emphasis is placed on the identification and use of strategies

across and within activities. Furthermore, the designers see the curriculum as stimulating social interaction among students as they cooperate, collaborate, and teach eachother within the computer learning environment. Creative Play encourages free exploration and discovery learning of problem solving activities on the computer.

Table 1 provides a brief description of each of the 25 programs. The curriculum is organized specifically on a rotating schedule so that students are introduced to a variety of programs with varying cognitive demands across the eight days of instruction. Table 2 provides an example of how the programs' cognitive skills are distributed across instructional sessions. It is apparent that Creative Play should activate a broad range of skills since each task demands several rather than a single cognitive activity. Not only do the programs stimulate interdependent cognitive activities but each instructional session requires a variety of those skills.

Most of Creative Play's first day of instruction focuses on language skills. Although the activities are categorized as logical reasoning, creative thinking, pattern recognition, and social interaction exercises, the tasks are heavily weighted toward verbal ability. Performance here depends on vocabulary, writing skills, and general facility with language. Day 2 emphasizes number skills as the programs rely on logical reasoning, data collection and evaluation, and pattern recognition. The games deal primarily with elementary rules of mathematics, the number system, and strategies that can be applied on logical reasoning/mathematics oriented exercises. The third day's activities add spatial visualization skills to those of logical reasoning and data collection and evaluation. The tasks all involve knowledge of the coordinate system. The programs in the fifth and sixth days of instruction are content-oriented simulations in the areas of geography, simple physics, and earth sciences. The final two days concentrate on spatial visualization skills for the maze exercises and pattern identification for the drawing programs. It is obvious from this brief description that Creative Play stimulates a variety of general problem solving and logical reasoning skills in a number of different contexts and content areas. The curriculum should prove to be an interesting test of the cognitive demands and consequences of learning in such a computer environment.

Study B-4: Yellow Light

Program Title:	Yellow Light
Requirements:	Apple II, Applesoft, One disk drive or Commodore PET (16K)
Description:	Simulation of physical concepts such as deceleration and acceleration
Color:	Yes
Sound:	Silent
Graphics:	High Res.
Grade Level:	High School

Table 1

Description of Programs

DAY	PROGRAM	DESCRIPTION
1	HANGMAN	Guess a vocabulary word in English.
	GEORGE	Guess a vocabulary word in Spanish.
	BIGRED	Help the computer write a story based on "Little Red Riding Hood."
	PASSWORD	Guess a word from synonym clues.
	WORDS	Deduce an English word from letter clues.
	SNAP	Recognize and react to words that are flashed on the screen.
	WFF	Use letters of the alphabet in a logic game. (This can be a difficult game for younger children.)
2	GUESS	Guess a secret number from one to one hundred.
	CUPS	Bet on which cup the computer will choose. The computer uses a pattern to choose successive cups.
	FUNCTION MACHINE	Put numbers into the Function Machine and observe the values that come out. Try to guess the mathematical rule that the machine uses.
	BRIDGES	Try to make the last move in a strategy game.
	NIM	Try to force your opponent to take the last pebble in a strategy game.
	STARS	Guess a number from one to one hundred using proportionality clues.

3	TICTAC	Play a coordinate version of tic-tac-toe.
	HURKLE	Find the Hurkle by guessing coordinates and receiving directional clues.
	MUGWUMP	Find the Mugwump by guessing coordinates and receiving distance clues.
4	LEM	Land a space capsule on the moon by controlling the ship's speed.
	RACING	Enter a car race and use laws of physics to safely speed around the track.
	CAVES	Explore a cave and try to find a path to fresh air and sunlight.
	BIRDS1	Investigate geography from a bird's perspective.
5	WUMPUS	Venture through a dodecahedron in search of the Wumpus. Shoot golden arrows and flee from super bats.
	CAVES	Explore a cave and try to find a path to fresh air and sunlight.
6	RATMAZE	Travel blindly through a computer-generated maze to find a piece of cheese.
	MAZE	Run a maze several times and record changes in your speed.
7	SYMMETRY	Draw patterns on the computer screen and then have the computer use these patterns to create symmetrical drawings.
	CHASE	Run from robots in a strategy game.

Table 2

Content Chart

DAY	Logical Reasoning	Data Collection & Evaluation	Creative Thinking
1	HANGMAN GEORGE PASSWORD WORDS WFF		Name Game The Number Situation BIGRED What is a Computer?
2	GUESS FUNCTION MACHINE BRIDGES NIM STARS	Investigating CUPS	
3	Coordinate Map TICTAC HURKLE MUGWUMP	HURKLE MUGWUMP	
4	Secret Message LEM RACING	CAVES	
5	CAVES WUMPUS	CAVES WUMPUS	Computer Guts
6	RATMAZE MAZE	MAZE	
7	CHASE Computer Talk		SYMMETRY
8			In a World of Computers. . .

Spatial Visualization	Pattern Identification	Social Interaction	Knowledge About Computers
	HANGMAN GEORGE SNAP	Name Game The Number Situation What is a Computer?	
	CUPS FUNCTION MACHINE NIM Investigating CUPS		
HURKLE MUGWUMP			
CAVES BIRDS1	LEM		Secret Message M&M Machine Computer Simulation
CAVES WUMPUS	WUMPUS		Computer Guts
RATMAZE MAZE			
SYMMETRY	SYMMETRY CHASE Computer Talk		Computer Talk
		In a World of Computers. . .	

Accompanying Materials:	Instructions for operations and program description. Teacher's guide with performance objectives, rationale, instructional materials, strategies, and program documentaiton.
Management Features:	Menu driven. Separated into (a) human response time simulation, (b) determination of go, stop, and dilemma zones, and (c) printouts of summaries of inputs and output.
Other Features:	Can be used to supplement classroom instruction or as a self-contained program.
Subject Area:	High School Physics
Specific Topic:	Acceleration and deceleration in a simulation of driving and stoppng at a yellow/red light.
Prerequisite Skills and Activities:	Knowledge of physics; acceleration, deceleration, and speed.
Instructional Purpose of Program:	Enrichment or standard instruction.
Instructional Techniques:	Simulation
Appropriate Group Instructional Size:	Individual

Yellow Light is a simulation of the physical principles involved in driving an automobile. In particular, Yellow Light focuses on acceleration and deceleration as they relate to stopping and starting a car at a highway intersection. The program is separated into three parts that require both physical and intellectual activity. The first section concerns the measurement of reaction time. The second part focuses on the determination of the "stop," "go," and "dilemma" zones on the highway. These three zones will be described in detail in a later section of this document. The final section presents a tabular summary of the input values for the critical physical variables and the resulting values for the three zones.

In the first part of the program where reaction time is measured, the student assumes the role of a driver who is approaching a standard intersection. The student is required to press the "g" key as if it were a car's gas pedal to make the car accelerate. The car approaches an intersection where a second car waits at the crossroad. When the traffic signal changes from green to yellow, the driver must hit the space bar, as if it were the brake pedal in an attempt to stop the car. Apparently there are two distinct components underlying this part of the simulation. First, there is the cognitive component whereby the student must decide whether to stop or go. Second, there is the physical expression of and response to that decision. The cognitive component requires that the student analyze the vehicle's location with respect to the intersection when the light changes to yellow. He then must decide whether there is a chance to proceed and thereby cross the intersection safely. This is a "go" decision. The student presses the "g" key as if it were the accelerator and attempts to

navigate across the intersection. Conversely, if the student determines that he cannot cross the intersection safely, given the car's position at the time of the light change, he then must press the space bar or brake pedal to stop the vehicle. The decision to stop or go is cognitive in nature. However, the student still must express that decision in terms of a physical response. That response takes the form of a keystroke on the computer. Yet, the expression of the cognitive decision is not as straightforward as it may appear. Errors can occur in the translation between cognitive processing and physical expression. First, the student can express his decision correctly by hitting the appropriate key. He reacts by hitting the "g" when he intends to go and the space bar to stop. However, there are instances when, by accident, he will press the wrong key. Although he intends to stop, for some reason he presses the "g" key, continues to the intersection, and therefore collides with the oncoming vehicle. It is in this instance that the cognitive decision and the physical expression of that decision become confused. Apparently, errors of this nature can occur at both the cognitive and physical level of processing. The car will either stop safely at the light or will ram the oncoming car. The reaction time for a successful stop is recorded. Several attempts can be made and the average across trials computed. If the driver does not maneuver the intersection safely, he then must decide whether or not to try again.

Performance on this first section emphasizes physical rather than cognitive activity. This does not imply that reaction time is unrelated to cognitive performance, but rather that the physical response may be the more critical factor here. As a motorist drives with a foot on the accelerator, so too does the student at the keyboard. When the light changes color, the driver must recognize, process, and then react to that information. Recognition of the stimulus change is the first action while the physical response is the second requirement. The faster the response, the better the performance. If the driver fails to stop safely, he then must analyze the situation to determine where the response error happened and how it can be corrected during subsequent trials.

The second and third phases concentrate on the cognitive requirements of the program. The student must consider the physical parameters of the simulation. The first parameter is reaction time, calculated in seconds. The possible choices range from .8 to 1.2 seconds. The speed limit is considered. It can range from 15 to 55 miles per hour. Deceleration is calculated in feet per second and ranges from 10 to 16. The duration of the yellow light can vary from 2 to 6 seconds. The final parameter is the width of the intersection and ranges from 15 to 60 feet. All of these variables influence the degree to which the driver can pass through an intersection safely.

The value of these parameters affect three output zones. First, there is the "go" zone in which the driver must continue across the intersection because he is too far down the road to execute a safe stop. Second, there is the "dilemma" zone in which the driver is forced to make a decision whether to continue or attempt to stop. Applying the brakes in the dilemma zone sometimes results in an

accident and at other times, safety. It is this dilemma zone that is most interesting. Finally, there is the "stop" zone in which the student should be able to stop safely if the brake is depressed soon enough.

Once the student determines the parameters, an intersection is displayed on the screen. The program calculates the three zones and tells the driver the length of the dilemma zone. The student is asked if there are any parameters he wishes to change. Changes to the input values affect the subsequent display. Students therefore can explore the cause and effect relationship between the input and output values. The display provides a visual representation of the intersection, showing the three zones. The third section of the program presents a tabular summary of the parameters' input values and the resulting zones. In this way, the student can analyze either a visual or numerical representation of how changes in the variables influence the output values.

The student can change the parameters in the program's second section as many times as desired. Following each change, the student is asked whether or not he would like to make any further changes. Following the last change, the student returns to the main menu. He then can examine the tabular results of all the combinations of input and output values.

The student must have a basic understanding of how physical concepts such as speed, acceleration, and deceleration interact with reaction time, the width of an intersection, and yellow light duration. Examination of the causal relationships among these parameters allows the student to gain an understanding of a few simple laws of physics. In addition to the knowledge gained within the specific content area of physics, the student also is required to use basic scientific inquiry skills. The manipulation of various parameters in a systematic manner is essentially the same process as testing a sequence of hypotheses. At first, the parameter changes may be thoroughly haphazard, but through experience, the student should be able to identify how the manipulation of one variable affects the output values of the three zones.

Consequently, Yellow Light is not simply a simulation within the physics content area. Rather, it also activates the use of logical reasoning, problem solving, and scientific inquiry skills. These skills include alertness to the stimuli, monitoring the causal relationships among variables, connecting new to old information, and planning subsequent parameter values and their resulting outcomes. The student must be able to monitor past trials to know what should be attempted in the future. Furthermore, he must analyze the results and isolate how specific changes in the parameters affect the values of the three zones. Such problem solving is done through activation of reasoning and inquiry skills, in combination with a more general knowledge of the content area. Yellow Light requires only an elementary understanding of the physical concepts. It can be used as supplementary material in the classroom or as a self-contained instructional program. The strength of Yellow Light is as a simulation

that stimulates the acquisition of higher cognitive skills in an area of physics where actual experimentation with the concepts is impractical and impossible. Students can gain an understanding of the presented concepts in the context of an enjoyable and educational simulation of the real world.

Study B-5: Pattern Discovery

Program Title:	Pattern Discovery
Requirements:	Apple II, Applesoft, One disk drive
Description:	Electronic canvas to explore solution of spatial problems.
Color:	Yes
Sound:	No
Graphics:	High Res
Grade Level:	Elementary and up.
Accompanying Materials:	Nine programs run from Applesoft. Keyboard input is automatic; no return stroke is necessary. Data can be collected on disk for inspection and analysis.
Other Features:	Can be used to supplement classroom instruction or as a self-contained program in art, math, or science curricula.
Subject Areas:	Scientific reasoning, spatial skills, and art.
Specific Topic:	Presentation of spatial problems in the context of abstract art to present mathematical and scientific reasoning skills.
Prerequisite Skills and Activities:	
Instructional Purpose of Program:	Training of spatial and reasoning skills, in particular to overcome sex bias in these skills.
Instructional Techniques:	Simulation and discovery learning.
Appropriate Group Instructional Size:	Individual, small group, or class.

Pattern Discovery is a simulation to teach scientific reasoning and spatial skills within the context and under the guise of abstract art. The context of art was selected as an attempt to engage girls and overcome the sex-bias found in typical existing spatial tasks.

The program is divided into nine separate subprograms of increasing difficulty. After booting a program, an initial canvas of spatial objects is displayed on the screen. The design consists of such variables as color, size, shape, and density. A typical canvas contains several lines of varying lengths, boxes and squares of varying sizes that are either open or solid, a background color, and different colors for the various geometric objects. Students then

are instructed to input three numbers, ranging from one to nine. Each number controls a particular variable displayed in the configuration. Sometimes extraneous variables are included in the more difficult problems to extend the students' ability to reason scientifically, isolate variables, and test hypotheses. The students' task is to determine which key controls which variable.

The purpose of the initial canvas is not to provide a baseline value at zero, but rather to provide the students with an idea of the spatial objects they are likely to see in the program. Once into the problem solving section of the program, students first must identify what variables in the form of spatial objects are being presented on the screen. Then, through a sequence of key strokes, they must attempt to determine the cause and effect relationship between key stroke and outcome of the configuration on the screen.

Since each of the three key strokes per individual sequence controls a particular variable, the students must eventually come to realize that there are optimal approaches to the task. Perhaps the most efficient strategy is to isolate one variable at a time. This is accomplished by varying one number while holding constant the other two numbers. For instance, say that the first digit controls the variable background color. A first sequence to isolate the variable would be 1-1-1 to obtain a baseline. The next sequence perhaps would be 9-1-1 to examine the extreme outcome for that variable. In the simpler programs, it may be sufficient to try this once or twice (including a mid-point such as 5-1-1) to determine if what is changing is background color and no other variables. However, in the more advanced programs, certain extraneous variables may be introduced to confound the students' perceptions of what is happening on the screen. In this instance, it is beneficial to repeat certain sequences to see what remains constant or changes, given the identical input from the keyboard. There may even be constants other than the variable of interest. After one variable is isolated, the students then can use a similar procedure by which to determine the two other variables.

There are several nice instructional features to this program. First, it is engaging to boys and girls alike. The use of art as the content area masks any potential sex discrimination evident in other forms of spatial tasks. Second, the programs are adaptable to any size instructional group from individuals to entire classes. Third, the programs can be integrated into a number of existing content areas (mathematics, science, or art), can stand alone as instructional materials or combined with ongoing curricula, and can be adapted to a variety of complexity levels. The response format required by the program minimizes students' typing burdens, allowing students to concentrate on their answers rather than the mode response. A response consists of only three keystrokes or numbers typed by the students. No return keystroke is necessary. Additionally, these keystrokes are saved in a data file for later access and can be dumped to a mainframe computer for subsequent statistical analysis. Finally and perhaps most importantly, Pattern Recognition teaches scientific reasoning and spatial skills that generally are not part of the explicit curriculum at any grade level. Students

can acquire an understanding of these concepts in a visually stimulating and cognitively engaging environment.

CHAPTER FOUR

Implications for Research

The previous chapters have covered a number of issues critical to the understanding of the cognitive demands and consequences of computer-assisted learning. In the first chapter, we outlined some of the claims that have been made recently about the effects of computer-assisted instruction on both individual learners and on the organization of learning environments in schools. We then described six characteristics of computer environments for learning that are expected to mediate learning in computer-assisted instruction. The first chapter concludes with a brief presentation of theories of learning and theories of pedagogy and comments on the role of these theories in the development of software. These issues reflect the political, social, psychological, and educational ramifications of bringing computers into classrooms and homes. In the second chapter, we examined the research relevant to computer environments for learning. Several bodies of literature were reviewed. The literatures on individual differences, instructional psychology, problem solving, planning, and CAI bring their own research agendas and paradigms to bear on the problem. The present review attempts to provide a comprehensive and integrated perspective on the variables likely to influence learning in computer environments for learning. In the third chapter, we described a taxonomy of software and discussed the combinations of characteristics of computer environments for learning that are expected to operate within specific classes in the taxonomy. We outlined several dimensions of software that may influence learning outcomes and concluded with a brief description of five pieces of software that embody characteristics hypothesized to foster the acquisition of higher order cognitive skills.

We now turn to several possible implications for future research on the effects of learning from computer software. Our intention in conducting this review has been to clarify issues relating to the design of our empirical research on the cognitive consequences of computer software. The cognitive consequences that we are most interested in are those that are generally included under the rubric of problem solving skills. Before commenting on the implications of the review, a few observations on the conduct of research on computing software are in order.

First, we must choose software for study at a time when the "half life" of educational software is relatively short. New software is being introduced into the educational market at a relatively fast pace. At least some pieces of software are likely to be obsolete after a very short implementation period. In order for research on software to be useful to software designers and educators, we must choose for study those aspects of software which are substantial. That is, we should

not expend resources on aspects of software that are likely to be remedied without the benefit of knowledge produced by the research community or are educationally uninteresting.

Second, the rate of change in computer hardware still seems to be somewhat greater than the rate of improvement in software. Computer hardware continues to change in physical size, cost, portability, and processing power. The changes are in the direction of smaller size, greater portability, more power, and lower cost. All of these factors affect the utility of computers for both school and non-school applications. Presumably these likely changes in computer hardware will contribute to continued change in the types and quality of educational software. As a result, we can expect continued rapid change in the technical or hardware related constraints on design and implementation of cognitively demanding software. To the extent that the role of research on educational software is to provide knowledge about the effects of particular aspects of the software, it would appear that such research will be directed at a moving target.

Third, the use of educational software in schools (and other settings as well) may place some constraints on the effects of that software on learners. As new and interesting software is introduced into schools, the initial uses of the software may not take advantage of its full potential. Substantial amounts of time and training are necessary before innovative educational software can be integrated into the school curriculum. For example, although word processing is being taught relatively frequently on microcomputers, word processing skills are (at this time) infrequently incorporated in an optimum manner into instruction on writing. That is, there is an emphasis on the use of word processor "functions" per se rather than on the use of word processing as a tool for composition. This example raises serious issues for researchers. Should we examine the effects of educational software as it is typically used now or should software be examined under some set of optimal conditions? Are we to be satisfied with an exploration of typical effects or of maximum effects?

These points illustrate a few of the contextual factors that bear on the conduct of research on educational software. These contextual factors constrain the research questions and methodologies that are currently feasible. Given this context, what factors are amenable to empirical research within the time and resources constraints of the ACCCEL project and, of those factors, which have the higher priority for solution? Our primary concern is to examine what is learned directly from interaction with specific examples of cognitively demanding educational software. Do students achieve the skills and knowledge that are directly involved in the activity itself? Beyond these direct outcomes, can students apply knowledge and skills acquired during interaction with cognitively demanding educational software to tasks that have not been previously encountered?

In terms of the conditions of instruction, we are primarily interested in different degrees of explicitness. By explicitness we mean the degree to which the tasks require cognitive processing on

the part of the learner. More versus less explicitness can be implemented by prescribing certain instructor behaviors (or "software interventions") when students are working with specific pieces of software. A second instructional condition that is considered of high priority is learner autonomy in relation to task definition and duration. This aspect of instruction can be varied considerably within both school and non-school settings. In conjunction with conditions of instruction, it is of high priority to examine learners with different characteristics. At this stage of the research, the most important of these characteristics are ability, cognitive style and gender.

These are the main themes that we recommend building into our studies of cognitively demanding educational software. This agenda does not preclude the consideration of other social, instructional, attitudinal, or descriptive factors in one or more substudies to be conducted by ACCCEL.

REFERENCES

Ali, A. M. (1981). The use of positive and negative examples during instruction: Some important issues related to the design and development of instructional materials. Journal of Instructional Development, 5, 2-7.

Anderson, R. C. (1967). Educational psychology. Annual Review of Psychology, 19, 129-163.

Atkinson, R. C. (1972). Ingredients for a theory of instruction, American Psychologist, 27, 921-931.

Ausburn, L. J., & Ausburn, F. B. (1978). Cognitive styles: Some implications for instructional design. Educational Communication & Technology Journal, 26, 337-354.

Baldwin, E. A., Liao, T. T., Corcoran, D., Reese, P., & Zengage, B. (1982). Yellow Light Problem. Department of Technology and Society State University of New York at Stony Brook.

Baron, J. (1981). Reflective thinking as a goal of education. Intelligence, 5, 291-309.

Barr, A., & Feigenbaum, E. A. (Eds.). (1982). The handbook of artificial intelligence (Vol. II). Los Altos, CA: William Kaufman.

Bredemeier, M. E. & Greenblat, C. S. (1981). The educational effectiveness of simulation games: A synthesis of findings. Simulation & Games, 12, 307-332.

Brickman, P., Linsenmeier, J. A. W., & McCareins, A. G. (1976). Performance enhancement by relevant success and irrelevant failure. Journal of Personality and Social Psychology, 33, 149-160.

Brown, A. L. & DeLoache, J. S. (1978). Skills, plans, and self-regulation. In R. Siegler (Ed.), Children's thinking: What develops? Hillsdale, NJ: Erlbaum.

Brown, J. S. In T. W. Malone & J. Levin (Eds.). (1981, March). Microcomputers in education: Cognitive and social design principles. Carnegie Corporation Conference held at the University of California, San Diego.

Brown, J. S., & Burton, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. Cognitive Science, 2, 155-192.

Brown, J. S., Collins, A., & Harris, G. (1978). Artificial intelligence and learning strategies. In H. F. O'Neil, Jr., (Ed.), Learning strategies. New York: Academic Press.

Bruner, J. S. (1973). Beyond the information given. New York: Norton.

Burbules, N., & Reese, P. (1983). The Rocky's Boots Study. In Research designs for studies of acquisition of higher cognitive skills from computer assisted learning (Draft Tech. Rep. 8), ACCCEL Project, Lawrence Hall of Science, University of California, Berkeley, CA.

Burton, R. R. (1981). Diagnosing bugs in a simple procedureal skill. Cognitive and instructional sciences series (Technical Report CIS-8). Xerox Palo Alto Research Center.

Burton, R. R., & Brown, J. S. (1979). An investigation of computer coaching for informal learning activities. International Journal of Man-Machine Studies, 11, 5-24.

Carr, B. (1977). WUSOR II: A computer aided instruction program with student modelling capabilities (AI Memo 417). Cambridge, MA: MIT.

Carrier, C. A., & Clark, R. E. (1978). Effects of presentation modes, explicitness, and student aptitudes on learning. Educational Communication & Technology Journal, 26, 329-336.

Cattell, R. B. (1963). Theory of fluid and crystallized intelligence: A critical experiment. Journal of Educational Psychology, 54, 1-22.

Cattell, R. B. (1971). Abilities: Their structure, growth, and action. Boston: Houghton Mifflin.

Center for Social Organization of Schools. (1983, June). School uses of microcomputers: Report from a national survey (Issue No. 2). Baltimore, MD: Author.

Cohen, P. R., & Feigenbaum, E. A. (1982). The handbook of artificial intelligence (Vol. 3). Los Altos, CA: William Kaufman.

Collins, A., & Stevens, A. L. (1981). A cognitive theory of interactive thinking. In C. M. Reigcluth (Ed.), Instructional design theories and models: An overview. New York: Academic Press.

Collins, A., & Stevens, A. L. (1982). Goals and strategies of inquiry teachers. In R. Glaser (Ed.), Advances in instructional psychology (Vol. 2), Hillsdale, NJ: Erlbaum.

Collins, A., Warnock, E. H., Aiello, N., & Miller, M. L. (1975). Reasoning from incomplete knowledge. In D. G. Bobrow and A. Collins (Eds.), Representation and understanding: Studies in cognitive science. New York: Academic Press.

Corno L., & Collins, K. (1981). Out of the starting gate: A system for enhancing the academic motivation of remedial students. Summer motivation study conducted in cooperation with the California State Department of Education, Office of Program Evaluation and Research and the Oakland Unified School District.

Corno, L., Collins, K. M., & Capper, J. (1982, March). Where there's a way there's a will: Self-regulating the low-achieving student. Paper presented at the annual meeting of the American Educational Research Association, New York.

Corno, L. & Mandinach, E. B. (1983). The role of cognitive engagement in classroom learning and motivation. Educational Psychologist.

Cox, B. A. H. (1980). Early adolescent use of selected problem-solving skills using microcomputers. Unpublished doctoral dissertation, University of Michigan, Ann Arbor, MI.

Cronbach, L. J. (1966). The logic of experiments on discovery. In L. S. Shulman & E. R. Keislar (Eds.), Learning by discovery: A critical appraisal. Chicago: Rand McNally.

Cronbach, L. J. & Snow, R. E. (1977). Aptitudes and instructional methods: A handbook for research on interactions. New York: Irvington Publishers.

Dalbey, J., & Linn, M. (1983). The demands and requirements soft computer programming: A review of literature (Tech. Rep. No. 1). Berkeley, CA: University of California, Lawrence Hall of Science.

Dewey, J. (1933). How we think: A restatement of the relation of reflective thinking to the educational process. Boston: Heath.

Diener, C. I., & Dweck, C. S. (1978). An analysis of learned helplessness: Continuous changes in performance, strategy, and achievement cognitions following failure. Journal of Personality and Social Psychology, 36, 451-462.

diSessa, A. (1981, March). In T. W. Malone, & J. Levin (Eds.), Microcomputers in education: Cognitive and social design principles. Carnegie Corporation Conference held at the University of California, San Diego.

Dweck, C. S. (1975). The role of expectations and attributions in the alleviation of learned helplessness. Journal of Personality and Social Psychology, 31, 674-685.

Dweck, C. S. (1976). Children's interpretation of evaluative feedback: The effect of social cues on learned helplessness. Merrill-Palmer Quarterly, 22, 105-109.

Dweck, C. S., & Bush, E. S. (1976). Sex differences in learned helplessness: I. Differential debilitation with peer and adult evaluators. Developmental Psychology, 12, 147-156.

Dweck, C. S., Davidson, W., Nelson, S., & Enna, B. (1978). Sex differences in learned helplessness: II. The classroom and III. An experimental analysis. Developmental Psychology, 14, 268-276.

Dweck, C. S., & Gilliard, D. (1975). Expectancy statements as determinants of reactions to failure: Sex differences in persistence and expectancy change. Journal of Personality and Social Psychology, 32, 1077-1084.

Dweck, C. S., & Reppucci, N. D. (1973). Learned helplessness and reinforcement responsibility in children. Journal of Personality and Social Psychology, 25, 109-116.

Eastman, C. M. (1970). Problem solving strategies in design (Reprint No. 23). Carnegie-Mellon University, School of Urban and Public Affairs.

Egan, D. E., & Greeno, J. G. (1973). Acquiring cognitive structure by discovery and rule learning. Journal of Educational Psychology, 64, 85-97.

Egan, D. E., & Greeno, J. G. (1974). Theory of rule induction: Knowledge acquired in concept learning, serial pattern learning, and problem solving. In L. W. Gregg (Ed.), Knolwedge and cognition. Potomac, MD: Erlbaum.

Flavell; J. H. (1976). Metacognitive aspects of problem solving. In L. B. Resnick (Ed.), The nature of intelligence. Hillsdale, NJ: Erlbaum.

Friedman, B., & Slesnick, T. (1982). Creative play: Problem solving activities with the computer. Math and Computer Education Project, Lawrence Hall of Science, University of California, Berkeley, CA.

Gagne, R. M., & Briggs, L. J. (1979). Principles of instructional design (2nd Ed.). New York: Holt, Rinehart and Winston.

Glaser, R. (1966). Variables in discovery learning. In L. S. Shulman & E. R. Keislar (Eds.), Learning by discovery: A critical appraisal. Chicago: Rand McNally.

Glaser R. (1982). Instructional psychology: Past, present, and future. American Psychologist, 37, 292-305.

Goldstein, I. (1976). The computer as coach: An athletic paradigm for intellectual education (AI Memo 389). Cambridge: MIT.

Goldstein, I. (1979). The genetic graph: A representation for the evolution of procedural knowledge. International Journal of Man-Machine Studies, 11, 51-77.

Goldstein, I. (1980). Developing a computational representation for problem-solving skills. In D. T. Tuma & F. Reif (Eds.), Problem solving and education: Issues in teaching and research. Hillsdale, NJ: Erlbaum.

Goldstein, I., & Carr, B. (1977). The computer as coach: An athletic paradigm for intellectual education. Proceedings of 1977 Annual Conference of the Association for Computing Machinery, Seattle, 227-233.

Goldstein, I., & Papert, S. (1977). Artificial intelligence, language, and the study of knowledge. Cognitive Science, 1, 84-123.

Gray, L. E. (1982). Aptitude constructs, learning processes, and achievement. Unpublished doctgoral dissertation, Stanford University, Stanford, CA.

Greeno, J. G. (1972). The structure of memory and the process of solving problems (Tech. Rep. 37). Ann Arbor: University of Michigan, Human Performance Center.

Greeno, J. G. (1976). Indefinite goals in well-structured problems. Psychological Review, 83, 479-491.

Greeno, J. G. (1978). A study of problem solving. In R. Glaser (Ed.), Advances in instructional psychology (Vol. 1). Hillsdale, NJ: Erlbaum.

Greeno, J. G. (1980). Some examples of cognitive task analysis with instructional implications. In R. E. Snow, P- A. Federico, & W. E. Montague (Eds.), Aptitude, learning, and instruction (Vol. 2). Hillsdale, NJ: Erlbaum, (a).

Greeno, J. G. (1980). Trends in the theory of knowledge for problem solving. In D. T. Tuma & F. Reif (Eds.), Problem solving and education: Issues in teaching and research. Hillsdale, NJ: Erlbaum, (b).

Greeno, J. G., Magone, M. E., & Chaiklin, S. (1979). Theory of constructions and set in problem solving. Memory and Cognition, 7, 445-461.

Hall, K. A. (1978). Computer-based education: Research, theory, and development. Educational Communication & Technology Journal, 26, 79-93.

Hayes, J. R. (1976). It's the thought that counts: New approaches to educational theory. In D. Klahr (Ed.), Cognition and instruction. Hillsdale, NJ: Erlbaum.

Hayes-Roth, B. (1980). Human planning processes (Tech. Rep. R-2670-ONR). Santa Monica: The Rand Corporation.

Hayes-Roth, B., & Hayes-Roth, F. (1978). Cognitive processes in planning (Tech. Rep. R-2366-ONR). Santa Monica: The Rand Corporation.

Hayes-Roth, B., & Hayes-Roth F. (1979). A cognitive model of planning. Cognitive Science, 3, 275-310.

Hayes-Roth, F. (1978). Learning by example. In A. M. Lesgold, J. W. Pellegrino, S. D. Fokkema, & R. Glaser (Eds.), Cognitive psychology and instruction. New York: Plenum.

Hayes-Roth & McDermott, J. (1976). Learning structured patterns from examples. Proceedings of the Third International Joint Conference on Pattern Recognition.

Hess, R. D., & Miura, I. (1983, April 25). Gender and socioeconomic differences in enrollment in computer camps and classes. Infoworld, 5, 52-55.

Humphreys, L. D., & Parsons, C. K. (1979). Piagetian tasks measure intelligence and intelligence tests assess cognitive development: A reanalysis. Intelligence, 3, 369-382.

Hunt, E. B., Martin, J., & Stone, P. J. (1966). Experiments in induction. New York: Academic Press.

Inhelder, B., & Piaget, J. (1958). The growth of logical thinking from childhood to adolescence. New York: Basic Books.

Johnson, E. S. (1971). Objective identification of strategy on a selection concept learning task. Journal of Experimental Psychology Monograph, 90, 167-196.

Johnson, E. S. (1978). Validation of concept-learing strategies. Journal of Experimental Psychology: General, 107, 237-266.

Kagan, J. (1966). Learning, attention and the issue of discovery. In L. S. Shulman & E. R. Keislar (Eds.), Learning by discovery: A critical appraisal. Chicago: Rand McNally.

Kahn, T. M. (1981). An analysis of strategic thinking using a computer-based game. Unpublished doctoral dissertaion, University fo California, Berkeley, Berkeley, CA.

Kiesler, S., Sproull, L., & Eccles, J. S. (1983). Second-class citizens? Psychology Today, 17, 40-48.

Larkin, J. H. (1980). Teaching problem solving in physics: The psychological laboratory and the practical classroom. In D. T. Tuma & F. Reif (Eds.), Problem solving in education: Issues in teaching and research. Hillsdale, NJ: Erlbaum.

The Learning Company. (1982). Rocky's Boots. Protola Valley, CA: Author

Lepper, M. R. (1982, August). Microcomputers in education: Motivational and social issues. Invited address presented at the annual meeting of the American Psychological Association, Washington, DC.

Lepper, M. R., & Malone, T. W. (1983, May). Intrinsic motivation and instructional effectiveness in computer-based education. Paper presented at the ONR Conference on Aptitude, Learning, and Instruction: Conative and Affective Process Analysis, Stanford University.

Liao, T. T. (1983). Using computer simulations to integrate learning. Simulation & Games, 14, 21-28.

Linn, M. C. (1978). Future research in adolescent reasoning, Final Report. National Institute of Education, Contract # NIE-P-78-0023.

Linn, M. C. (in press). Theoretical and practical significance of formal reasoning. Journal of Research in Science Teaching.

Linn, M. C., & Siegel, H. (in press). Post-formal reasoning: A progressing research program. In M. Comms (Ed.), Beyond formal operation: Late adolescent and adult cognitive development. New York: Praeger.

Linn, M. C., Fisher, C. W., Mandanich, E. B., Dalbey, J., & Beckum, L. (1982). Demands and cognitive consequences of computer learning. Proposal submitted to the National Institute of Education.

Linn, M. C., Pulos, S., & Gans, A. (1981). Correlates of formal reasoning: Content and problem effects. Journal of Research in Science Teaching, 18, 435-447.

Machiavelli, N. (1961). The prince (G. Bull, trans.). Great Britian: Penguin Books.

Malone, T. W. (1980). What makes things fun to learn? A study of intrinsically motivating computer games (Tech. Rep. CIA-7). Xerox Palo Alto Research Center.

Malone, T. W. (1981). Toward a theory of intrinsically motivating instruction. Cognitive Science, 5, 333-369.

Malone, T. W., & Lepper, M. R. (1983, May). A taxonomy of intrinsic motivation. Paper presented at the ONR Conference on Aptitude, Learning, and Instruction: Conative and Affective Process Analysis, Stanford University.

Malone, T. W., & Leving, J. (Eds.). Microcomputers in education: Cognitive and social design principles. Carnegie Corporation Conference held at the University of California, San Diego.

Mandinach, E. B. (1983). The WUMPUS study. In research designs for studies of acquisition of higher cognitive skills from computer assited learning (Draft Tech. Rep. 8), ACCCEL Project, Lawrence Hall of Science, University of California, Berkeley, CA.

Mandinach, E. B. (In progress). Clarifying the "A" in CAI for learners of different abilities. Unpublished doctoral dissertation, Stanford University, Stanford, CA.

Martorano, S. C. (1977). A developmental analysis with performance on Piaget's formal operations task. Developmental Psychology, 13, 666-672.

Mayer, R. E. (1974). Acquisition process and resilience under varying testing conditions for structurally different problem-solving procedures. Journal of Educational Psychology, 66, 644-656.

Mayer, R. E. (1975). Different problem-solving competencies established in learning computer programming with and without meaningful models. Journal of Educational Psychology, 67, 725-734 (a).

Mayer, R. E. (1975). Forward transfer of different reading strategies evoked by testlike events in mathematics text. Journal of Educational Psychology, 67, 165-169 (b).

Mayer, R. E. (1981). The psychology of how novices learn computer programming. Computing Surveys, 13, 721-141.

Mayer, R. E. & Greeno, J. G. (1972). Structural differences between learning outcomes produced by different instructional methods. Journal of Educational Psychology, 63, 165-173.

McCann, P. H. (1981). Learning strategies and computer-based instruction. Computers and Education, 5, 133-140.

McDermott, D. (1978). Planning and acting. Cognitive Science, 2, 71-109.

McKeachie, W. J. (1974). Instructional psychology. Annual Review of Psychology, 25, 161-193.

Merrill, M. D. (1980). Learner control in computer-based learning. Computers and Education, 4, 77-95.

Messick, S. (1982, March). Cognitive styles in educational practice. Paper presented at the annual meeting of the American Educational Research Association, New York.

Messick, S. (1983, May). The correlational structure of intelligence, personality, and cognitive style. Paper presented at the ONR Conference on Aptitude, Learning, and Instruction: Conative and Affective Process Analysis, Stanford University.

Miller, G. A., Galanter, E., & Pribram, K. H. (1960). Plans and the structure of behavior. New York: Holt, Rinehart and Winston.

Miller J. (Chair) (1983, April). Computer-based instruction: A future oriented tutorial. Invited symposium at the annual meeting of the American Educational Research Association, Montreal.

Nicholls, J. G. (1975). Causal attributions and other achievement-related cognitions: Effects of task outcome, attainment values, and sex. Journal of Personality and Social Psychology, 31, 379-389.

Orbach, E. (1977). Some theoretical considerations in the evaluation of instructional simulation games. Simulation & Games, 8, 341-360.

Orbach, E. (1979). Simulation games and motivation for learning: A theoretical framework. Simulation & Games, 10, 3-40.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. New York: Basic Books.

Parsons, J. E., Ruble, D. N., Hodges, K. L., & Small, A. W. (1976). Cognitive-developmental factors in emerging sex differences in achievement-related expectancies. Journal of Social Issues, 32, 47-61.

Pellegrino, J. W., & Glaser, R. (1982). Analyzing aptitudes for learning: Inductive reasoning. In R. Glaser (Ed.), Advances in instructional psychology (Vol. 2). Hillsdale, NJ: Erlbaum.

Polya, G. (1954). Induction and analogy in mathematics: Vol. 1: Of mathematics and plausible reasoning. Princeton, NJ: Princeton University Press.

Polya, G. (1957). How to solve it (2nd ed.). Princeton, NJ: Princeton University Press.

Pulos, S., & Linn, M. C. (1981). Generality of the controlling variable scheme in early adolescence. Journal of Early Adolescence, 1, 26-37.

Reder, L. M. (1980). The role of elaboration in the comprehension and rentention of prose: A critical review. Review of Educational Research, 50, 5-54.

Resnick, L. B. & Glaser, R. (1976). Problem solving and intelligence. In L. B. Resnick (Ed.), The nature of intelligence. Hillsdale, NJ: Erlbaum.

Rowher, W. D., Jr. (1980). An elaborative conception of learner differences. In R. E. Snow, P-A. Federico, & W. E. Montague (Eds.), Aptitude, learning, and instruction (Vol. 2). Hillsdale, NJ: Erlbaum.

Royer, J. M. (1979). Theories of the transfer of learning. Educational Psychologist, 13, 53-69.

Ruble, D. N., Parsons, J. E., & Ross, J. (1976). Self-evaluative responses of children in an achievement setting. Child Development, 47, 990-997.

Salomon, G. (1974). Internalization of filmic schematic operations in interaction with learners' aptitudes. Journal of Educational Psychologist, 14, 53-69.

Salomon, G. (1979). Media and symbol systems as related to cognitiion and teaching. Journal of Educational Psychology, 71, 131-148.

Schank, R. (1980). How much intelligence is there in artificial intelligence? Intelligence, 4, 1-14.

Seigner, R. (1980). Game ability and academic ability: Dependence on SES and psychological mediators. Simulations & Games, 11, 403-421.

Shulman, L. A., & Keislar, E. R. (Eds.). (1966). Learning by discovery: A critical appraisal. Chicago: Rand McNally.

Simon, H. A., & Lea, G. (1974). Problem solving and rule induction: A unified view. In L. W. Gregg (Ed.), Knowledge and cognition. Potomac, MD: Erlbaum.

Slavin, R. E. (1980). Cooperative learning. Review of Educational Research, 50, 315-343.

Snow, R. E. (1976). Research on aptitudes: A progress report. Technical Report 1, Aptitude Research Project, School of Education, Stanford University (a).

Snow, R. E. (1976). Research on aptitude for learning: A progress report. In L. S. Shulman (Ed.), Review of Research in Education, 4, 50-105 (b).

Snow, R. E. (1980). Aptitude and achievement. New Directions for Testing and Measurement, 5, 39-60 (a).

Snow, R. E. (1980). Aptitude, learner control, and adaptive instruction. Educational Psychologist, 15, 151-158 (b).

Snow, R. E. (1980). Aptitude processes. In R. E. Snow, P-A. Federico, & W. E. Montague (Eds.), Aptitude, learning, and instruction (Vol. 1) Hillsdale, NJ: Erlbaum (c).

Snow, R. E. (1982). Training of intellectual aptitude. In D. K. Detterman & R. J. Sternberg (Eds.), How and how much can intelligence be increased. Norwood, NJ: Ablex.

Stansfield, J. L., Carr, B. P., & Goldstein, I. P. (1976). Wumpus Advisor I: A first implementation of a program that tutors logical and probabilistic reasoning skills (AI Memo 381). Cambridge: MIT.

Sternberg, R. J. (1977). Intelligence, information processing, and analogical reasoning: The componential analysis of human abilities. Hillsdale, MJ: Erlbaum

Sternberg, R. J. (1980). Sketch of a componential subtheory of human intelligence. Behavioral and Brain Sciences, 3, 573-584.

Sternberg, R. J. (1981). Intelligence and nonentrenchment. Journal of Educational Psychology, 73, 1-16.

Sternberg, R. J. (1982). A componential approach to intellectual development. In R. J. Sternberg (Ed.), Advances in the psychology of human intelligence (Vol. 1). Hillsdale, NJ: Erlbaum.

Sternberg, R. J. (1983, May). Intelligence, conation, and affect. Paper presented at the ONR Conference on Aptitude, Learning, and Instruction: Conative and Affective Process Analysis, Stanford University.

Sternberg, R. J., & Davidson, J. E. (1982). The mind of the puzzler. Psychology Today, 15, 37-44.

Sternberg, R. J., & Davidson, J. E. (1983). Insight and the gifted. Intelligence, 18, 51-57.

Teske, J. A., & Pea, R. D. (1981). Metatheoretical issues in cognitive science. The Journal of Mind and Behavior, 2, 123-178.

Thorndike, E. L. (1913). Educational psychology: Vol II: The psychology of learning. New York: Columbia University Press.

Tobias, S. (1982). When do instructional methods make a difference? Educational Researcher, 11, 4-9.

VanLehn, K., & Brown, J. S. (1980). Planning nets: A representation for formalizing analogies and semantic models of procedural skills. In R. E. Snow, P-A. Federico, & W. E. Montague (Eds.), Aptitude, learning, and instruction (Vol. 2). Hillsdale, NJ: Erlbaum.

von Oech, R. (1982). A whack on the side of the head. Menlo Park, CA: Creative Think.

Weitheimer, M. (1959). Productive thinking. New York: Harper.

Wilensky, R. (1981). Meta-planning: Representing and using knowledge about planning in problem solving and natural language understanding. Cognitive Science, 5, 197-233.

Wilensky, R. (1983). Planning and understanding. Reading, MA: Addison-Wesley.

Witkin, H. A. (1949). The nature and importnace of individual differences in perception. Journal of Personality, 1, 145-170.

Wittrock, M. C. (1966). The learning of discovery hypothesis. In L. S. Shulman & E. R. Keislar (Eds.), Learning by discovery: A critical appraisal. Chicago: Rand McNally.

Yalow, E. (1980). Individual differences in learning from verbal and figural materials (Tech. Rep. 12). Aptitude Research Project, School of Educationi, Stanford University.

Yinger, R. J. (1977). A study of teacher planning: Description and theory development using ethnographic and information processing methods. Unpublished doctoral dissertation, Michigan State University.

Yob, G. (1975). Hunt the WUMPUS. Creative Computing, 2, 51-54.